Smooth Ramps: Walking the Straight and Narrow Path through Color Space.

Conference Paper · January 1999

Source: DBLP

CITATIONS
9

READS
373

1 author:

Thor Olson
University of Minnesota Twin Cities
6 PUBLICATIONS **22** CITATIONS

SEE PROFILE

Smooth Ramps: Walking the Straight and Narrow Path through Color Space

Thor Olson
Management Graphics Inc
Minneapolis MN

Abstract

Smoothly shaded color ramps, important for presentation graphics and computer imaging, are difficult for color managed systems utilizing device profiles. The causes for disruptive artifacts such as contours and banding are examined and a set of visual limits are established that, if met will avoid them. Based on these visual thresholds, an analysis of the numeric representation and processing of color data in an ICC profile-based environment yields some requirements for device profiles and their use.

Specifically, we find that 8-bit precision and inadequate table indexing resolution cause contour artifacts. Banding is caused indirectly from the inversion of noisy and nonlinear printer color data when the profile was created. The noise is not instrument noise, but rather due to the inconsistency of printer output. Some proposals for improving this stage of profile making are suggested. Examples are provided to illustrate the sources of difficulties in rendering smooth and uniform color ramps.

Keywords:
color encoding, color management, color precision, color processing, differential color, discrimination, luminance ramp, presentation graphics, ICC profile, visual artifacts, visual tolerances,

Introduction and motivation

A smoothly varying ramp between two different colors is a frequently encountered object in computer graphics. It is used in presentation graphics to make pleasing backgrounds and it appears in natural or synthetic scenes, wherever the lighting falls off gradually over a colored object.

With more than half of digitally printed color pages originating from RGB software and intended for presentation [BER99], the ability to render color ramps smoothly and without visual artifacts has become quite important. Yet current research topics in color imaging tend to be oriented toward absolute color measurements and errors rather than the relative, or differential, color behavior of systems.

To illustrate the distinction, it has been observed that "dumb" RGB to CMYK color conversions can render a smoothly graded color ramp, albeit perhaps in the wrong hue, while a more accurate, profile-based conversion results in a coarse ramp with obvious banding (figure 1). Why should this be? This paper will examine such artifacts, determine their causes, and describe the requirements to avoid them.

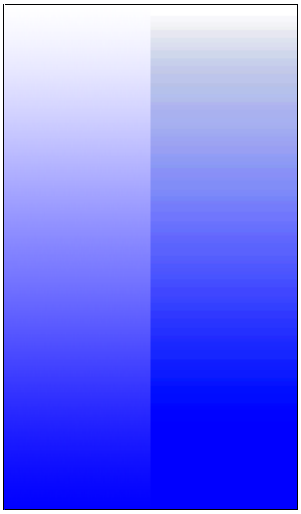

Figure 1. This blue to white ramp, popular for presentations, is rendered using a simplistic RGB to CMYK conversion (left) and using color management (right). The color-managed ramp may be more hue-accurate, but the other is smooth and uniform. The color signals have been amplified by 2X for illustration purposes. The actual output shows these qualitative aspects at a diminished (but still highly visible) level.

Figure 2. Color ramps depend on the space in which their endpoints are specified. Here are two distinctly different ramps from blue to green: RGB (top) and CIELAB (bottom).

Definitions
A color ramp is defined by its endpoints. Two distinct colors are placed, and the colors between them are uniformly distributed along their connecting line. A digital color ramp has endpoint colors that are quantized in the space they reside in, and the intervening colors are constructed as discrete steps.

A *smooth* ramp is one that contains no steps. This is obviously a visual condition, since a digital ramp is comprised entirely of steps. The visually smooth ramp contains no visually discernable steps.

A *uniform* ramp properly (linearly) interpolates its endpoints. It contains steps that are equally spaced in the ramp's source color space and it is *straight*; containing no "bumps" that take it to the side of its line. Again, this is a visual assessment, since the digital interpolation will in general not land on the exact line it is interpolating.

Note that uniformity depends on the ramp's color space (figure 2). It will be visually straight and uniform only if specified in a visually uniform space. This is certain to not be the case; and the visual result can only be as straight and uniform as the space being interpolated. To make any headway, we accept this as a larger issue, and take as our metric, distances in CIELAB, which while globally not uniform, is locally uniform and widely used for expressing color differences, the very issue we care about. We can then consider differences from a target line in CIELAB and leave it to other parts of the system to bring the color space into a larger scale visual uniformity (see for example, [BF98], [MAR98]).

As a framework to explore the numerical requirements for representing ramps, we consider a profile-based color management system. A sample task is to take a ramp specified in RGB and convert it to the CMYK space of a printer. The way in which this is done is to convert the RGB to a common *profile connection space* (PCS, either CIEXYZ or CIELAB), and then convert from there to the desired output space (figure 3). The conversions are accomplished through the use of device profiles conforming to the ICC specifications used widely in the industry.

To scrutinize the processing involved, we examine the case where the RGB to PCS conversion step occurred perfectly. The issues we will uncover apply to both sides of the PCS, but we start with an idealized representation of our ramp within PCS.

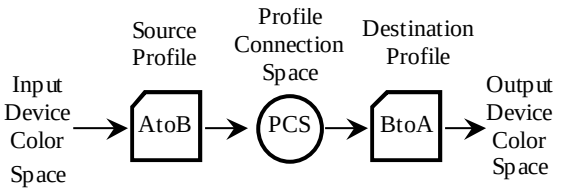

Figure 3. Schematic of color management system using device profiles. The profiles are essentially data structures that represent the relationship between device space and PCS.

The most common way to represent the printer color space is through the use of the capable combination of table structures shown in figure 4. It is a three-stage structure, the first and last being direct conventional lookup tables that implement 1D functions. The middle stage is a 3D table which, to conserve memory, is sparse: a table with a coarse grid of data which are interpolated to obtain the result. For reasons of performance, this interpolation is a linear operation on the immediate neighbor grid points, and the front- and back-end 1D luts are direct lookups (no interpolation between table entries).

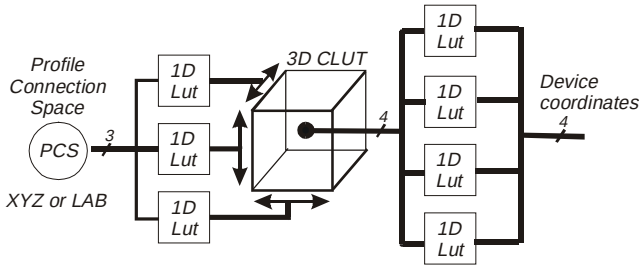

Figure 4. The ICC output profile structure for a 4-colorant device such as a CMYK printer.

Visual Tolerances

If we study the ramps of figure 1, we find that there are two main types of artifacts sometimes called "contours" and "banding". There are sudden discrete steps in the ramp, and there are broader bands of erroneous color. We will obtain two visual tolerances that apply to their visibility.

What causes contours?
It is commonly stated in the literature that the discrimination of the human visual system (a just-noticeable-difference) is represented by a ΔE distance measurement in CIELAB of approximately one. This may be true in some viewing environments, but we have consistently found that this is too large to use as a step size for creating visually smooth ramps. This is easily demonstrated by trying to construct a

black to white ramp using 100 steps of L*. In all but the most limited displays, the step artifacts will be readily apparent.

Other ramps along a* or b* do not show as much sensitivity, and our experiences in making colored ramps indicate that this type of artifact is dominated by the luminance channel. In fact, earlier studies suggest that in a low noise, high dynamic range image no fewer than 400 steps are required in order to guarantee visual smoothness [NIDL91, OLS92]. This explains the difficulties experienced by users of 8-bit software tools in creating visually smooth gradients; there are not enough codes available for the number of steps required.

Other researchers investigating visual thresholds have found peak sensitivities consistent with this finding; a summary of some of them is shown in table 1.

Table 1. Some findings on human visual sensitivity to luminance steps and sin wave gratings.

Reference	ΔL/L	Approx ΔL*
VNB65, 525nm, luminous display	0.0025 bp	0.15
BB94 Reflection prints	0.0025 bp	0.15
PR70 Video display	0.002-0.005 rms	0.2 – 0.5
PC Steps in gratings	0.0075 pp	0.25
NIDL91 Video display	$.0166\ (L/L_0)^{0.828}$ (0.008 pp @ 50 ft-L)	0.25
OLS92 Backlit film	$0.01 + 0.001\ \ln^2(L/L_0)$	0.25

The apparent contradiction with much industry practice is due to the reduced requirements of many imaging applications. When only complex images are used, or the dynamic range is limited, or noise is added from sources such as halftoning and film grain, the number of required steps is reduced. Image noise is an important factor, but luminance gratings can be seen even when they are drowned in wideband noise that is 75 times larger [PR70]. We have found it necessary to deliberately add noise to 8-bit images (up to 5-bits worth, 32 counts) when the output device does not have adequate step control. The human visual system appears to be able to pull out a signal even when the signal to noise ratio is –18dB!

Since we are attempting to find the limiting case where such masking effects are not present, we do not consider it excessive (and perhaps not aggressive enough) to take as our smoothness requirement a maximum step size of ΔL*of 0.25.

Although units for the a* and b* scales of CIELAB are intended to provide the same "strength" as L* to indicate color differences, this seems to be approximately true only at low spatial frequencies. If we conduct a comparative experiment where we make ramps that traverse larger distances using a fixed number of steps, the ramps in a* and b* can go considerably further than L* before step artifacts are noticeable (figure 5).

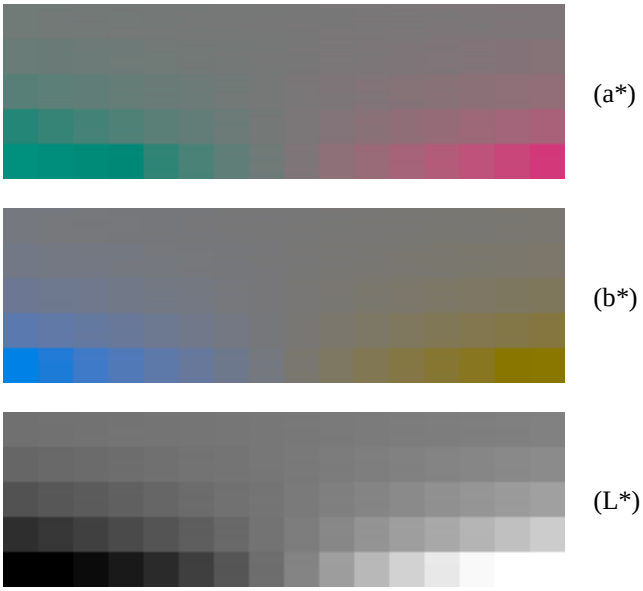

Figure 5. Step increments in CIELAB. The top row of each set increments by 0.5 units per step, followed by 1, 2, 4, and 8 units per step. This reproduction may not discern the finest steps but at the row where L steps become visible, the corresponding a* and b* rows will still seem smooth, indicating that a given number of steps can smoothly span a larger distance in a* and b* than in L*.*

Clearly the sensitivity to the high frequency step artifacts is less in the a* and b* axes, and we can increase the tolerable step size along them. However, if we want to make ramps with arbitrary endpoints and utilize a mathematically isotropic color difference space, we must use the more restrictive distance as a maximum tolerable ΔE in the steps of a color ramp.

What causes banding?

A phenomenon of slightly lower bandwidth shows up as apparent deviations from the intended color within a ramp. In understanding this and other details of color ramps, we will first subtract the "signal" (the intended ramp) and study what remains, the "noise". We then want to know whether

this noise is visible or not. We justify this step by noting that when we study concepts of smoothness and uniformity, we are not very interested in the ramp itself, but in the higher frequency components of it. Visually, this is spatial adaptation, not seeing the ramp but sensing the deviations from it. Mathematically, this is a subtraction of one of the endpoints, and its interpolation. We end up with a ramp with the same endpoint colors (not much of a ramp!) and then examine the steps in between.

The tool we can use to analyze the residual is the visual contrast sensitivity function (figure 6). Again we find higher sensitivity to variations in luminance than chrominance. Our highest sensitivity occurs at about 6 cycles per degree where we can sense less than 0.2% contrast [COR70].

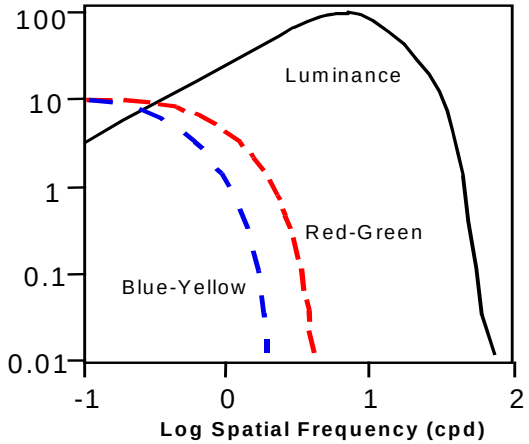

Figure 6. Contrast sensitivity function for luminance and color signals (adapted from [FAI98]).

We can derive a useful limit for working with banding artifacts by noting that the contrast sensitivity follows a line that falls by about a factor of 5 for every decade drop in frequency [DAV68]. This can be expressed as:

$$T(f) = A \cdot 5^{-Log(f)} \qquad (1)$$

where A is the amplitude of the threshold at 1 cycle per degree. We will use A=0.03 $\Delta L_{pp}/L$, a fairly liberal number consistent with [PR70]. For our later purposes, we would like to obtain a threshold estimate that represents our sensitivity to triangle waves. We first find the triangle wave that has this sin amplitude as its fundamental. This requires a modest scaling by $\pi^2/8$. We next note that the threshold data were measured as a modulation of the adapted white level. To convert to ΔL^*_{pp} we scale by the slope of L* at white (116/3). Our final value for A is 1.5 ΔL^*pp, the peak-to-peak amplitude of a 1 cyc/deg triangle wave at the threshold of visibility.

Our sensitivity to banding artifacts is related to both the slope and the amplitude of the luminance error. Rapid changes in luminance are tolerable, provided they are limited to a short interval, and hence small maximum error. Larger total errors are tolerable, but must be spread over a longer baseline, meaning that the error *gradient* must be limited. A plot of maximum error against the minimum distance over which it must be distributed is shown in figure 7. The ratio of maximum error to minimum distance establishes a limiting gradient.

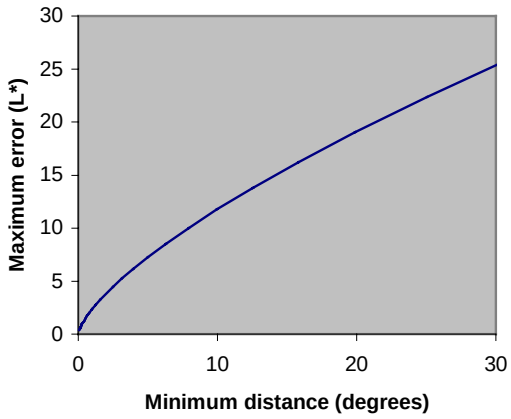

Figure 7. A maximum L error is plotted against the minimum angular distance it must span in order to not be visible.*

If we are to make straight ramps, the errors must fall below the threshold and its gradient must be restricted. A way to determine this is to evaluate the gradient (of the error) along the ramp and to track the maximum amplitude of the error. If, at any point, the slope exceeds the maximum allowed error gradient for that error level, we are in trouble.

A relation exists to provide a simple check:

$$f \cdot T^{1+Log(e)} \cong k \qquad (2)$$

The threshold amplitude raised to the 1.43 power, then scaled by its frequency is a constant (approximately, to within 1%). For our L* and cyc/deg units, the value for k is 1.7. Using this formula, we can quickly find the baseline over which an error must be spread:

$$\lambda = \frac{T^{1+Log(e)}}{k} \qquad (3)$$

We will use this relation later in estimating requirements for device profiles.

Any proposal for visual tolerances is dependent on the application and the environment where such tolerances are

used. Our numbers suggest that the visual system is more sensitive than many in this industry would deem appropriate or necessary, but they are consistent with other investigations into accuracy requirements. This is perhaps peculiar to our specific business, but the important point here is that estimates for visual tolerances can be made, and then used to determine the numerical requirements of a color system.

Numerical Tolerances

To be useful in digital imaging, the visual tolerance numbers set forth in the previous section must be translated into the numerical tolerances used for the encoding and processing of color. The following topics will illustrate how to determine these numerical tolerances. In the process we will find out some of the requirements for the data structures contained in the ICC color profile, our sample processing environment.

Smoothness – PCS quantization

We established that to maintain the appearance of being smooth, the steps in a digital ramp could not increment by more than 0.25 units in L*. The first structure we encounter in the profile are the independent (1D) lookup tables. Regardless of whether the profile connection space (PCS) is CIELAB or CIEXYZ, we are forced at this point to discretize the space to perform the lookup. If the PCS is CIELAB, this means that the lut must have at least 800 entries (to avoid adjacent round-up and round-down errors exceeding 0.25). If the PCS is XYZ, we find that 7,200 entries are required! This is a consequence of the linearity of XYZ and its high sensitivity near black. (Most ICC profiles do not utilize luts this large, in fact the largest possible is 4096).

The contents of the 1D luts are used as an index into the interpolated 3D color lookup table (clut). We know that at least 400 steps are needed to span the largest possible ramp, presumably from black to white. This means that the contents of the 1D luts must be at least 9 bits in size. The ICC format allows 8 or 16-bit entries, obviously 8-bit profiles are not suited to making smooth ramps

The purpose of the 1D luts is to convert the PCS into a space which is "linear" with respect to the 3D clut, so that its sparse representation can be linearly interpolated. We don't know the exact space that the clut is in, this is a choice made by the profile creator, but it is frequently indexed by functions that are quite close to (if not exactly), L*, a*, b*. In the following, we consider the case where it actually is so, and that the front-end 1D luts successfully convert the PCS to CIELAB without error.

Smoothness – device control

We can think of the clut as holding calibrated device coordinates that generate the desired color. At this stage of the analysis we pretend that the device has a continuous, infinite precision control, and concern ourselves only with the effects of the linear interpolations of a sparse sampling of this control.

If the device were calibrated and behaved as a CIELAB device, there would be no error resulting from interpolating between the grid points in the clut (remember, our index into the table is also CIELAB). On the other hand, the clut would be superfluous; it implements no useful function, setting aside its use for gamut mapping. Since most devices do *not* use L*a*b* as their control signals, the clut is the key to making the transformation between coordinates. Problems in maintaining step increments will arise because of the mismatch of coordinates. We are linearly interpolating the contents of the clut, which is very likely to be *not* linearly related to our index.

It is easy to see the problem with a severe example. Say the device behaved and was calibrated as a linear luminance device. Further, we take an extreme case of a clut, one with only a single cell, which contains only the endpoints of the color space. If we take uniform steps along the L* axis in generating a ramp, we will be linearly interpolating the black and white endpoints (figure 8). Clearly the control signals obtained will be quite incorrect, but ignoring their lack of accuracy, they will make step to step increments that are highly visible!

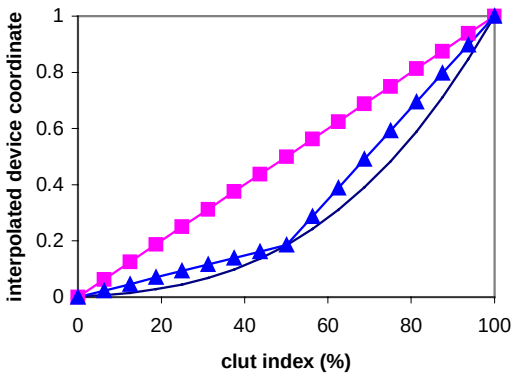

Figure 8. An extreme example illustrates including more gridpoints in the clut in order to better approximate the device control function. Wherever the slope *of the line segment is steeper than the function it is approximating, step artifacts are likely.*

The obvious solution is to use more grid points in the clut. Adding a single gridpoint in the middle improves the behavior considerably (figures 8 and 9). At the gridpoint, the output of the device is exactly correct. But wherever the slope of the linear approximation is greater than the slope of the control function, the steps will be "amplified". Note that it is not possible to solve the problem by just increasing the number of gridpoints, because in *any* interpolation interval there will be a region where the approximating line is steeper than the function. It can be solved everywhere only if the step interval along the index is reduced to *below* the visibility threshold so that there is some margin for its subsequent amplification.

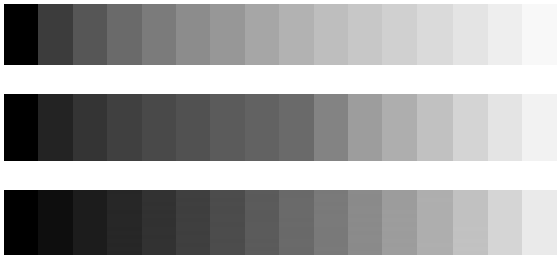

Figure 9. (a) a clut with no internal gridpoints, (b) shows the shift of step sizes by adding a single gridpoint. Compare adjacent step sizes to (c), the exact scale, and with the line segment slopes in figure 10.

We can quantify the requirement by defining a "step sensitivity" function S, the amplification factor between the clut index and the device output. It is the local gradient of the output (as expressed in the units of our uniform color space) with respect to the index of the clut, c. The expression for the step sensitivity S for the L* axis is:

$$S(c) = \frac{dL^*(c)}{dc} = \frac{d}{dp}L^*(p) \cdot \frac{d}{dc}p_{\text{int}erp}(c) \qquad (4)$$

L*(p) is the printer response, p_{interp} is the printer control code obtained by interpolating the clut. An interpolation interval corresponds to a cell in the clut and ranges from 0 to N; N+1 is the total number of gridpoints along an axis in the clut. Within interval n, the value obtained is:

$$p_{\text{int}erp}(c,n) = p(c_n) + \frac{c - c_n}{c_{n+1} - c_n} \cdot \left(p(c_{n+1}) - p(c_n)\right)$$
$$\text{for } c_n \le c \le c_{n+1} \qquad (5)$$

where c_n = n/N are the clut gridpoints and the printer control codes are specified there exactly as $p(c_n)$. Wherever the step sensitivity function exceeds unity, we must index the clut in smaller increments to avoid visible steps in the ramp. The reduction factor is the sensitivity value. So in our two-cell simple example, we must index the clut using steps that are a factor of 3 smaller than the L* visibility step, this factor being the peak value of its sensitivity function (figure 10).

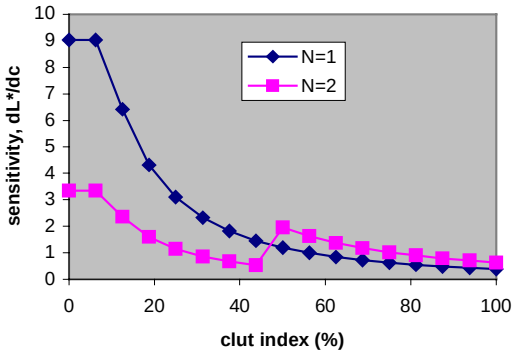

Figure 10. The step sensitivity function is the derivative of the output with respect to the clut index (L).*

While most printers do not have such a strong control function as this linear luminance example, they will all have sections that are affected in this way. As a more realistic example, consider an output device that is calibrated to follow a pure gamma 2.2 response. This is very similar to L*, but there are regions where the slopes are not matched. The sensitivity function depends on the relation of the device and the clut index and the number of gridpoints (figure 11). An interesting behavior in this example is that because of the mismatch between the linear section of L* and a pure 2.2 power law, the step sensitivity actually *increases* in the shadow regions when the number of gridpoints is increased!

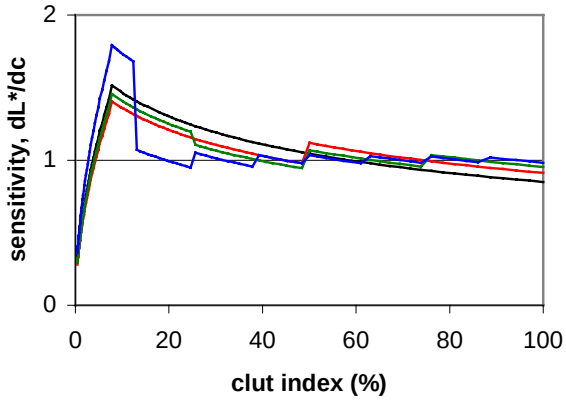

Figure 11. Step sensitivity function for an output device that behaves as a perfect gamma 2.2 power law. A family of curves shows gridpoint counts of N=1, 2, 4, 8. Note that adding more gridpoints does not always reduce the peak step sensitivity.

Smoothness – device quantization

We are now at the output stage of the color profile. A value for the device control level has been obtained by interpolating the 3D clut. The purpose of this last stage is to provide a calibration mechanism, between the actual device codes needed and the contents of the clut. In the case of the linear luminance example of the previous section, the strong device function should have been implemented here, leaving a more gentle response for the clut. This reduces the errors due to interpolating the sparse clut array, but places a resolution burden on this final output stage.

There is enough burden already. Just as the front-end lookup table required enough entries to guarantee sufficiently small steps, so too must the back-end. If the lut is implementing a device transfer function, there must be enough resolution so that successive entries do not make a visible step. Recall that because this is a *lookup*, there must be at least as many entries as there are visible steps across the color space (more than 400 along L*). If the table is indexed by any other function, the number of entries only increases.

Further, the precision of the lut *contents* must be adequately high. Devices that are better behaved than the linear luminance example will not need this full precision, but obviously to generate 400 distinct steps, more than 8-bits are required.

Uniformity

We move from analyzing the high frequency step effects to lower frequency tolerances. Here we are concerned with errors that take us off-course from the idealized line through color space that the ramp should follow. The source of these errors is primarily the 3D clut.

Color profiles are constructed by taking many output color measurements. Measuring all possible colors is impractical, so a sampling grid of 4 to 8 per axis is used yielding a few hundred to several thousand color samples. The contents of the profile's clut are obtained by "inverting" the measured data by sampling on a grid in PCS of typically 8 to 33 per axis, and finding the device coordinate that obtains that color.

If there were no noise, we could analyze the tolerances needed to perform the inversion of one sparse array into another. We would find, as in the previous section, that it depends on the functional behavior of the printer. If our printer was a perfect CIELAB printer, or a linear transform of it, we could invert it perfectly and there would be zero error in any entry of the clut.

But there *is* noise. One immediately thinks about the accuracy and repeatability of the measuring instrument, but it turns out that this is not a large factor [GHR97]. A good color-measuring instrument causes only about a 0.1 delta-E uncertainty. This is below the threshold of visibility, and because the error is spread out over the sampling grid interval, the differential error that we care about becomes negligible.

A much more significant factor is the repeatability and controllability of the printer itself. The variations in attempting to create a specific color from the printer control codes have been found to be in the range of 5 ΔE [GHR97]. Our own experiences with color copiers show that uniformity across the page and repeatability between pages is at least this bad. This means that the instrument is accurately measuring a color which only loosely represents a particular point in the printer's color space. Each point in the device grid is a sample within a "cloud" of possible colors that the device makes with that code. The diameter of the cloud is 5 ΔE.

This might not be so bad if there were local correlation between our samplings of these clouds, because for our purposes we are less concerned with absolute accuracy than with good differential behavior. There is some local correlation; immediately adjacent color codes show less uncertainty with respect to its neighbor. But we must accept that the sparse color patch measurements we make for building a color profile contain 2 to 3 delta-E of uncorrelated noise. Figure 12 shows a plane of measurements made in preparation for generating a profile showing the resulting kinks and slope variations.

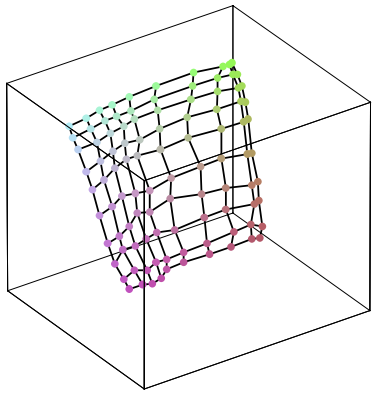

Figure 12. A leaf of printer data plotted in CIELAB.

What is the visibility of these kinks? If only a single measurement contained an error and we were making a ramp through it, we would encounter a visual bump. If the amplitude of the bump were 2 delta-E, we know from our triangle wave limit formula (equation 3) that it must be

spread over a minimum angular wavelength span of 1.5 degrees

If the error was from a clut that used an 8x8x8 grid, a full ramp must not be any smaller than 6 degrees to avoid this error from becoming visible. At a grid density of 16, the ramp can be no smaller than 12 degrees. By using a 32-grid, the ramp must no longer be printed on a letter-size page or it will be detected at reading distance! We find that increasing the grid density, while possibly providing more accuracy, makes banding artifacts more visible.

To illustrate, the contents of two actual printer profiles show the deviations of the output device codes as one traverses a section along L*. In the first case, the printer codes show an oscillation with respect to a straight line connecting the ends of the range. In the second, the maximum deviation from the straight line is about the same but the output codes vary smoothly.

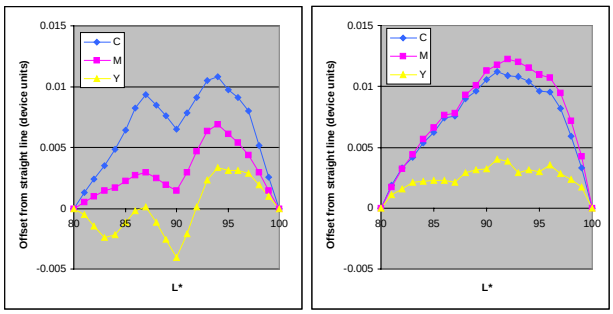

Figure 13. A trace along a section of L through two profiles plot the deviation from a straight line for the C, M, and Y printer codes. It is not significant that the codes deviate from a straight line, rather that the variations smoothly and correctly represent the printer response.*

Figure 14. A natural image processed through the two profiles of figure 13. Banding artifacts are apparent in the background of the left picture. These device codes have been amplified by 2x for illustration.

A natural image was rendered through each profile. Even though this is an image with photographic and digital noise, banding artifacts are readily seen in the background when it is rendered through the first profile. The improvement using the second profile is shown in figure 14. We conclude from this that the printer is, locally and on average, a smoothly behaved device but that any specific measurement of it will contain significant variations that can become embodied in its profile.

This appears to be the source of the broad bands and lumps in our color-managed ramps. How do we get rid of them? The problem is not the precision of the processing; it is the uncorrelated noise content of our original measurements (combined with possible nonlinearity and grid sampling effects). While this problem has *not* been solved to our satisfaction, we believe the following are reasonable approaches for its amelioration:

1. As proposed in [GHR97], average several patch measurements to reduce the space and time varying component of the noise. This is unpleasant in that it further increases the work to sample the color space of the printer.

2. Process the measurements in such a way that the random noise is removed by weighted averaging with neighbors, or constrained curve fitting. Unfortunately, the noise is wideband, and only the higher frequency components can be eliminated in this way.

3. Sample the color space more densely, use the additional samples to allow improved filtering. This combines both 1 and 2 above to both average and smooth the data. Like item 1, it increases the effort required to make a profile.

4. Filter the resulting profile clut contents. By processing the end product of the profile generation process, we trade off color accuracy for smoothness.

5. Generate a data set from a model, which takes as parameters the information extracted from the measurements.

6. Abandon the uniform grid, develop an alternate method to represent the color space of the printer which maintains sufficient accuracy and allows for better differential behavior and proper handling of the gamut edge. This is obviously a departure from the ICC architecture.

Of the above strategies, we have experimented with (2), using polynomial fits to the data, and (4) by various weighted averages of an entry in the clut with its neighbors. The beneficial results of removing high frequency components from the profile are seen in the example above.

Conclusions

Luminance errors are the dominant factor generating artifacts in smooth shading and digital ramps. To guarantee that you will never see them, the step size should be limited to delta-L* <= 0.25.

Unless you have considerable noise in your 8-bit imaging system, you will see step artifacts in digital ramps.

Smooth ramps cannot be generated using 8-bit precision ICC profiles. Sixteen-bit precision is necessary, and the 1D luts must have more than 256 entries.

Low frequency (banding) errors must be limited in both amplitude and gradient. A useful relation is based on the triangle wave visibility limit of 1.5ΔL*pp at 1 cpd. The threshold and gradient limits are related by a power law.

Printer noise is captured in a profile. The grid densities of the measurement set and the profile clut influence how the printer noise results in banding artifacts.

By careful attention to the precision of color processing and frequency content of a device profile, it is possible to make smooth and straight color ramps. We end the discussion by showing the introductory color ramp after it has been more carefully color-managed.

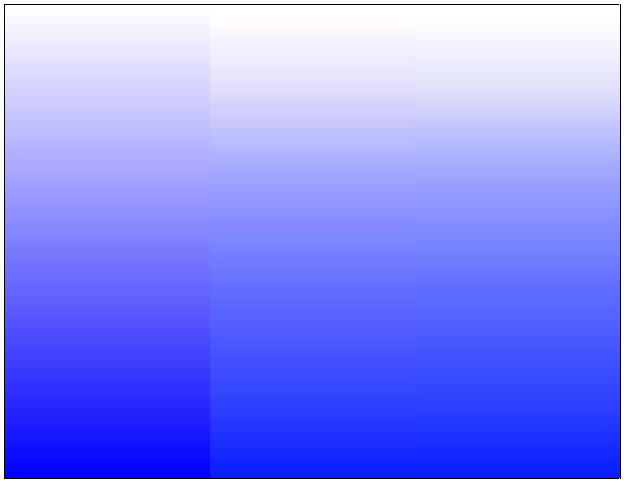

Figure 15. Left: simple RGB to CMY, center: color managed ramp with inadequate profile precision and excessive noise content, right: improved color processing. The color signals have been amplified by 2 for this illustration.

References

[BB94] Norman Burningham and Theodor Bouk, "Threshold Visibility and Objectionability of Banding in Reflection Prints", IS&Ts Tenth International Congress on Advances in Non-Impact Printing Technologies, 1994.

[BER99] Bergren, Kristin. Private communication regarding market useage of color copiers connected to digital sources, 1999.

[BF98] Gustav Braun and Mark Fairchild, "Color Gamut Mapping in a Hue-Linearized CIELAB Color Space", IS&T and SID Sixth Color Imaging Conference, Nov 1998.

[COR70] Tom N. Cornsweet, Visual Perception, Harcourt Brace Jovanovich, 1970.

[DAV68] M.L. Davidson, "Perturbation approach to spatial brightness interaction in human vision", J. Opt. Soc. Am. v58.

[FAI98] Mark D. Fairchild, Color Appearance Models, Addison-Wesley, 1998.

[GHR97] G. Gonzalez, T. Hecht, A. Ritzer, A. Paul, J.f. Le Nest, M. Has, "Color Management- how accurate need it be?", IS&T and SID Fifth Color Imaging Conference, Nov 1997.

[MAR98] Gabriel Marcu, "Gamut Mapping in Munsell Constant Hue Sections", IS&T and SID Sixth Color Imaging Conference, Nov 1998.

[NIDL91] National Information Display laboratory, "Monitor Evaluation and Video Specifications", Version 1.1, Princeton NJ, March 21, 1991.

[OLS92] Olson, Thor. "Digital Film Part 2: Good Grays and Continuous Colors", 134th SMPTE Technical Conference, Nov 1992.

[OLS96] Olson, Thor. "Deriving Digital Imaging Specifications: Some Requirements for Artifact-free Continuous Tone Output Part 2: Pixel Intensity Coding" IS&T 49th Conference, Minneapolis 1996.

[PR70] Herbert Pollehn and Hans Roehrig, "Effect of Noise on the Modulation Transfer Function of the Visual Channel", Journal of the Optical Society of America, v 60, n 6, June 1970.

[VNB65] F.L. Van Nes and M.A. Bouman. The effects of wavelength and luminance on visual modulation transfer. *Excerpta, Medica, Intern.Congr. Ser. No. 125*. "Performance of the Eye at Low Luminance." Proc. of the Symposium, Delf.t 1965.