Published as: R. Bala, V. Monga, G. Sharma, J. P. VandeCapelle, "Two-dimensional transforms for device color calibration", *Proc. SPIE: Color Imaging: Processing, Hard Copy, and Applications IX, vol. 5293,* San Jose, 19-22, Jan 2004, pp. 250-261.

Two-dimensional transforms for device color calibration

Raja Bala[1], Vishal Monga[2], Gaurav Sharma[3], Jean-Pierre Van de Capelle[4]

[1, 4]Xerox Corporation
800, Phillips Road, Webster, NY – 14580
[1]RBala@crt.xerox.com, [4]Jean-Pierre.van.de.Capelle@usa.xerox.com

[2]Dept. of Electrical and Computer Engineering
The University of Texas at Austin, Austin, TX – 78712-1084
vishal@ece.utexas.edu

[3]Dept. of Electrical and Computer Engineering
University of Rochester, Rochester, NY– 14627
gsharma@ece.rochester.edu

ABSTRACT

Color device calibration is traditionally performed using one-dimensional per-channel tone-response corrections (TRCs). While one-dimensional TRCs are attractive in view of their low implementation complexity and efficient real-time processing of color images, their use severely restricts the degree of control that can be exercised along various device axes. A typical example is that 1-D TRCs in a printer can be used to either ensure gray balance along the C = M = Y axis or to provide a linear response in ΔE units along each of the individual (C, M and Y) axis but not both. This paper proposes a novel two-dimensional calibration architecture for color device calibration that enables significantly more control over the device color gamut with a modest increase in implementation cost. Results show significant improvement in calibration accuracy and stability when compared to traditional 1-D calibration.

1. INTRODUCTION

Achieving consistent and high quality color reproduction in a color imaging system necessitates a comprehensive understanding of the color characteristics of the various devices in the system. This is done through a process of device calibration and characterization[1]. Device calibration is the process of maintaining the device with a fixed known characteristic color response and acts as a precursor to characterization. The characterization process establishes the relationship between device dependent signals and a device independent representation for the calibrated device. Calibration and characterization are key components in a color management system, enabling images to be communicated and exchanged in a device-independent color space and then transformed to device dependent signals prior to rendering[2]. In general, device independent color spaces are chosen based on the characteristics of human vision. To this end, color spaces such as CIELAB and CIEXYZ are popular choices for the device independent representation.[3 ,4,5]

This paper focuses on the calibration function. The process of calibrating the device to a specific color characteristic typically requires making color measurements and deriving correction functions to ensure that the device maintains that desired characteristic.[1] E.g. for a 3-color (CMY) printer, the goal of calibration is to determine a calibration transform from CMY (device values as determined by the characterization) to C'M'Y' (device values actually given to the printer) that maintains a desired printer response in selected regions of color space. Traditionally, the calibration transform that achieves this response is applied in the form of one-dimensional TRC correction to the individual channels of the device. Mathematically, the TRC mapping can be defined as $C' = f_1(C)$, $M' = f_2(M)$, $Y' = f_3(Y)$. A significant benefit in using 1-D TRCs is that these are very efficient for real-time processing of color images.

Examples of responses specified for 1-D calibration include: i) linearization of each channel to ΔE from paper; ii) linearization and gray-balance so that C=M=Y maps to the L* axis in CIELAB space. The device characteristics in those regions of CMY space outside of these controlled axes are an outcome of the calibration function, and hence cannot be controlled. For example, all input points with the same C value (an M-Y plane perpendicular to the C axis) will map to the same C' value. The result is that the response in a large part of color space is potentially compromised to achieve a desired response along a 1-D locus. If, for example, the TRCs are gray-balanced, this can sometimes result in undesirable hue shifts in the reproductions of sweeps from white to secondary device colors (for instance those lying along the blue axis where C=M, and C = K = 0). Conversely, if the TRCs are independently linearized, this will generally not yield gray balance for C = M = Y. As an alternative to control of each individual channel for calibration, the calibration may be performed as a full 3-D function from input C, M, Y to output C', M', Y'. Mathematically stated, $C' = f_1(C, M, Y)$, $M' = f_2(C, M, Y)$, $Y' = f_3(C, M, Y)$. With 3-D corrections, the desired response can be specified at any point in three-dimensional CMY space. Thus it is possible to derive unique one-to-one calibration functions without any compromise in the desired response. The implementation of the calibration transform would require 3-dimensional (3-D) look-up tables (LUTs). A full resolution 3-D LUT with direct lookup is prohibitively large. For 8 bit processing for instance, a full 3-D LUT would require $3*(256)^3$ bytes = 48MB of storage. Alternatively, sparse 3-D LUTs can be used in conjunction with some form of 3D interpolation.[6,7,8,9] In this case, the interpolation operation can be too computationally intensive for high speed printing applications. The number of measurements required to derive 3-D calibration would also be much larger than that required for 1-D calibration. The problem becomes even more involved with a 4-color CMYK printer.

In this paper, we propose a novel alternative, namely the use of two-dimensional (2-D) full-resolution lookup tables for device calibration.[10] The 2-D LUTs are expected to offer a compromise between 1-D and 3-D calibration both in terms of control and complexity. The additional dimension afforded by a 2-D LUT offers additional control, and hence significant improvements over what can be achieved with 1-D TRCs. At the same time, the memory and computational requirements for 2-D calibration, while higher than those for 1-D TRCs, are still low enough for incorporation into real-time high-speed printing applications. In addition, the required measurements for deriving the 2-D LUTs are also of the same order as traditional one-dimensional methods. The concepts developed for calibration can also be applied for device emulation, where the goal is to mimic the response of one device with another.

2. BACKGROUND

As mentioned earlier, the device independent color paradigm commonly adopted by color management systems requires the derivation of color transformations that map between device dependent color and device independent color. For output devices, this is normally accomplished with the following steps:

1) A set of color patches with pre-determined device control values is sent to the device and the color of each patch is measured in device-independent color coordinates.
2) The data from step (1) is used to derive a forward device-response function that maps device control values to the device independent values produced by the device in response to the control values. Techniques for estimating the forward function can be either model-based,[11, 12, 13] or empirical,[1] or a hybrid between the two.[14]
3) The forward device-response function is inverted to obtain a device-correction-function, which maps each device-independent color to the device control values that produce the specified device-independent color value on the output device. The device-correction-function is typically pre-computed and stored.
4) In order to produce a given color on the output device, the corresponding device-independent color values are mapped through the device correction-function to obtain control values. When the device is driven with these control values, the desired color is produced.

The rest of this discussion will focus on the specific case of printers. It is common practice to separate the device correction-function into two parts: a calibration function that immediately precedes the device; and a characterization function, which addresses the device through the calibration function. This separation is illustrated in Fig. 1 for the case of a CMYK printer. There are several advantages to this partitioning. First, the calibration function can be viewed as a preconditioning function that ensures that the device characteristics meet certain constraints, e.g. smoothness, monotonicity, etc. prior to derivation of the characterization. Secondly, temporal variations in the device's response, and variations within a fleet of devices can often be addressed in the calibration function. The concatenation of the calibration function and the device can be viewed as a "calibrated output device" that is stable in time and across the fleet and a single characterization can be used for the entire fleet and over time instead of requiring characterization of each device in the fleet over time.

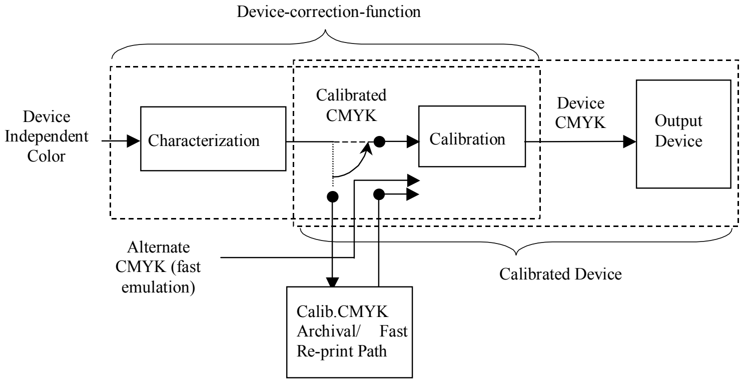


Figure 1. Schematic representation of the partitioning of the device-correction function into characterization and calibration.

Third, other settings that impact the device color response, for example the choice of the halftone scheme in a binary printer, may be compensated for to some extent in the calibration process. This allows for a common characterization function to be used for the different device settings; thus saving time, effort, and memory as the measurement intensive characterization does not need to be repeated for each setting. In addition to halftones, other examples of device settings include output print media and viewing illuminant. Some of these may be compensated for in the calibration process.[15]

Finally, there is another important benefit to the partitioning in Fig. 1. Recall that calibration involves specifying and maintaining a desired device response within selected regions of device color space. Examples of desired responses that have thus far been cited are channel-wise linearization and gray-balance. One can generalize this notion and specify the desired response as that of a reference output device. This is referred to as device emulation. The reference device to be emulated may be i) an actual device; ii) an industry standard device (e.g. SWOP offset press[16] or sRGB display[17]); or iii) an idealized device (e.g. a device that exhibits channel-wise linearity or perfect gray-balance). Some background on device emulation is in order, and will be presented next.

The goal in device emulation is to transform a set of device control values corresponding to the emulated device into the final control values for the emulating device such that the emulating device's response to these final control values is the same as the emulated device's response to the original device control values. This transformation is referred to herein as the emulation transform. The latter can be conceptually achieved by concatenating the emulated device's forward device-response function with the emulating device's device-correction function, as illustrated in Fig. 2. In general, the emulation transform is a computation and memory intensive multi-dimensional function. E.g. if the emulated and emulating devices are both CMYK printers, the emulation transform is a 4-D (CMYK to CMYK) transform. It is however possible to use the calibration transformation as a simpler alternative for providing approximate, or partial emulation. This is typically achieved by deriving the calibration transformation to coincide with the full emulation transform within one or more selected regions. These regions are typically the ones that are critical to color reproduction. Essentially, the problem of calibrating a device to achieve specific color characteristics can be restated as the problem of emulating a reference device that possesses those characteristics. For the rest of the paper, the following notation will be adopted:

1.) $\mathbf{v}_s$ - a vector of control values in the device space of the emulated device (e.g. CMY/CMYK)
2.) S_s - the complete set of control values in the device space of the emulated device (e.g. CMYK cube)

3.) g( ) - forward device response function for the emulated device (e.g. CMY → CIELAB mapping)
4.) $\mathbf{v_e}$ - a vector of control values in the device space of the emulating device (e.g. CMY/CMYK)
5.) y( ) - forward device response function for the emulating device (e.g. CMY→ CIELAB mapping)
6.) $\mathbf{r_s}$ - a vector of response values (e.g. CIELAB, or delta E from paper)

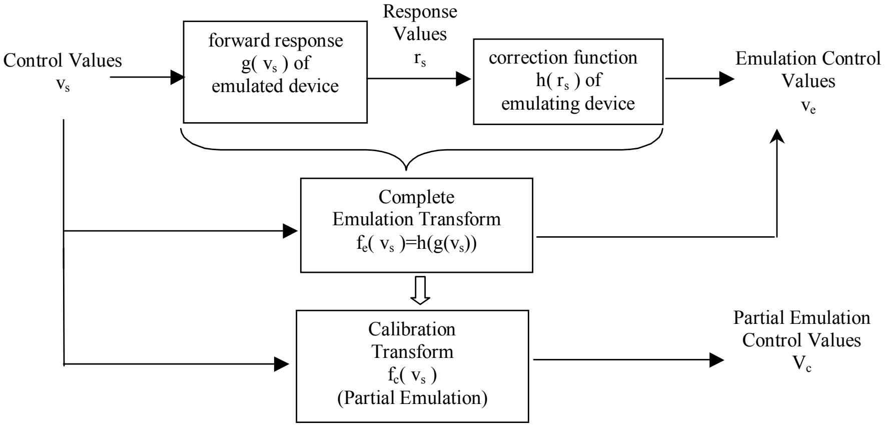


Figure 2. Emulation of one device on another through concatenation of emulated device's forward response and emulating device's device correction function.

7.) h( ) - the emulating device's correction function (e.g. mapping from CIELAB to printer CMY/CMYK)
8.) f_e( ) - full emulation transform from emulated device's control values to emulating device's values
9.) f_c( ) - calibration transformation
10.) S_e - the region in device space wherein calibration coincides with full emulation. This can be described mathematically as:

$$S_e = \{\mathbf{v}_s \in S_s: f_c(\mathbf{v}_s) = f_e(\mathbf{v}_s)\};\ S_e \subset S_s \qquad (1)$$

From the aforementioned discussion, it should be clear that the purpose of the calibration transformation is to facilitate a trade-off. Unlike the full device-correction function, the calibration transform provides control of the output device only in a limited fashion. However, in comparison to the full device-correction function it offers significant advantages in terms of reduced measurement effort and computational simplicity. The latter enables calibration to be executed in high-speed real-time printing applications.

3. One-Dimensional Calibration

Calibration traditionally involves one-dimensional correction applied to the individual channels for the device. For instance for a CMYK printer, the calibration is applied as tone-response corrections (TRCs) for each of the C, M, Y, and K channels. Typically, the specification of the response to be emulated consists of two parts. The first part is a 1-D response such as density or ΔE from paper for the K axis (C = M = Y = 0) and is realized by independent calibration of the K-axis. The second part of the specification specifies the desired response for the "CMY printer" obtained when the K-channel is turned off (K=0). The following discussion refers to the case of 3 color CMY calibration. The calibration architecture for a CMY printer is shown in Fig. 3. The two most common examples of 1-D calibration are described next.

3.1. Channelwise linearization to ΔE from paper

In this type of calibration, each channel is independently linearized to the ΔE_{ab} (deltaE) color difference between the i^{th} (i = C, M, Y etc.) channel and medium (paper) white. Described in the context of our emulation framework, the calibration process is attempting to emulate an ideal device whose channel-wise response is linear to ΔE_{ab} from paper. This response may be defined as

$$r_s = g(v_s) = \Delta E_{ab}(c_{medium}\ ,\ c(v_s)) = \frac{v_s}{v_s^{\max}}\left\{\Delta E_{ab}(c_{medium}\ ,\ c(v_s^{\max}))\right\} \qquad (3)$$

v_s = input control value for one of C, M, Y channels
v_s^{max} = maximum input control value (e.g. 255 for 8 bit inputs)
c_{medium} = CIELAB measurement of the medium
$c\ (v_s)$ = CIELAB response corresponding to input value v_s

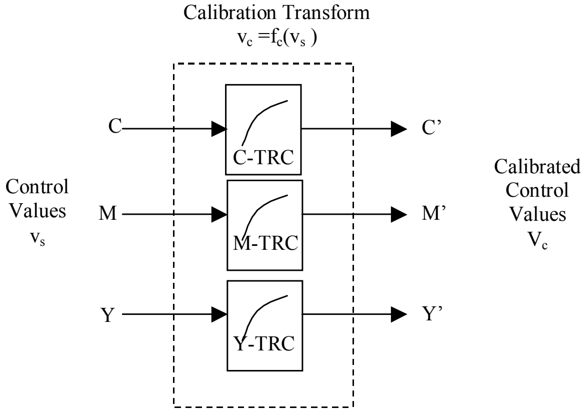


Figure 3. Traditional 3-Color 1-dimensional Calibration Transformation

Note, the emulated region S_e comprises the C, M, and Y axes, and a 1-D response is specified for each axis.

3.2. Gray-balanced calibration

For 3-color gray-balanced linear response, the response in CIELAB is specified along the device gray axis (C = M = Y), as the combined requirement that C = M = Y map to the L* axis in CIELAB and that equal steps along the C = M = Y axis correspond to equal steps in L*. (This can be readily generalized to an arbitrary response curve in L*.) In our emulation framework, the device being emulated is an ideal device that exhibits the characteristic that v_s = C = M = Y maps to the neutral (L*) axis. Determining the calibration functions $f_c(v_S)$ that engender this gray-balanced response involves searching for device C, M, Y that produce the desired neutral color on the emulating device. Many search techniques exist that can suitably address this problem.[1]

There are two major motivations for gray-balance calibration. First the human visual system is particularly sensitive to color differences near neutrals. Secondly, gray-balancing considers, to a first order, interactions between C, M and Y that are not taken into account in channel-wise calibration. Note that in practice, most printers cannot be gray-balanced beyond a level $v_S^{maxgray} < v_S^{max}$ where the TRC output v_c corresponding to one of the C, M, Y channels saturates prematurely to its maximum value. We may still require v_c for each channel to span the full range $[0, v_c^{max}]$. This implies that in the dark region the gray balance has to be compromised to allow each of the TRCs to go to its maximum control value (i.e. 255 in an 8 bit system). Furthermore this compromise must be arrived at smoothly in order to avoid discontinuities near the printer black point. Details can be found in Ref [1].

3.3. Performance analysis

As stated earlier, 1-D TRCs are very efficient for real-time image processing both in terms of computation and storage and memory requirements. E.g. for 8-bit processing, 256 bytes of memory are required for each channel of the TRC. For CMY printers, this amounts to 768 bytes of storage.

1-D TRCs however, offer severely limit the amount of control that can be exercised. Note that the specification of a desired response along the C = M = Y (device-gray axis) produces an uncontrolled response along the individual axes. Both cannot be independently specified with 1-D calibration. One example of an undesirable effect due to this coupling is sometimes seen as “hue” variations in secondary device-colors when the TRCs are determined for gray balance, for example e.g. a red ramp corresponding to C=M, Y=K=0 may undergo undesirable shifts with TRCs determined for gray balance along the C=M=Y axis. In general, one can see that a change in the “Cyan TRC” influences not only the corresponding point along the C=M=Y axis but also all other points in input CMY space having the same Cyan value. This idea is illustrated in Fig. 4. The specified response to be emulated for the point C_0=M_0=Y_0 on the device-gray axis determines the corresponding values for all three of the C, M and Y 1-D LUTs/TRCs shown in Fig. 3. However, the value of the Cyan TRC for the point C_0 influences the entire C=C_0 plane, the value of the Magenta TRC for the point M_0 influences the entire M=M_0 plane, and the value of the Yellow TRC for the point Y_0 influences the entire Y=Y_0 plane.

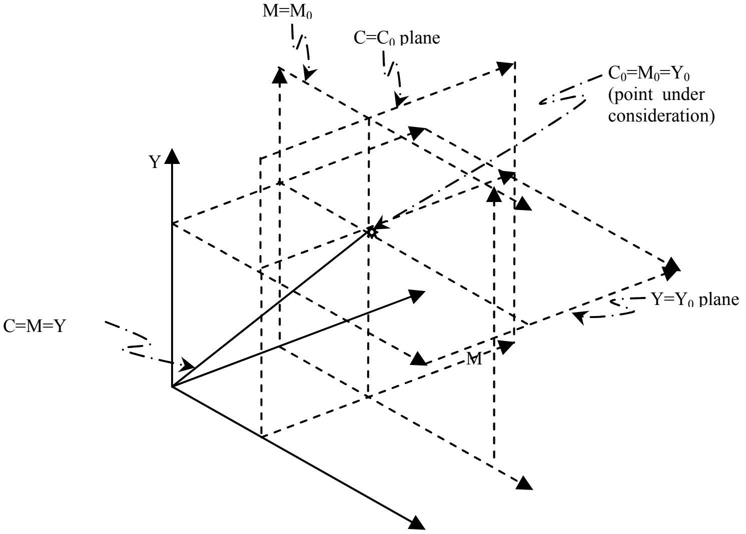


Figure 4. Illustration of influence of calibration controls outside the emulation set for 1-D TRCs

If the desired response for the "CMY 3-color printer" over the emulation set S_e is specified in 3-D CIELAB coordinates, the restrictions on the emulation set S_e imply for the specific case of the 1-D calibration that each point in the emulation set S_e map to distinct C values, distinct M values, and distinct Y values. In other words, each point where the response to be emulated is specified must differ from all other points in all three of the Cyan, Magenta, and Yellow control values (for the device to be emulated). By allowing compromises and by virtue of smoothness this requirement may be slightly relaxed in practice. The practical implication is that the 3-D response to be emulated may only be specified along a curve in CMY space that is monotonic in all three of the coordinates C, M and Y. This restriction is a result of the limited control offered by the traditional 1-D calibration.

4. TWO-DIMENSIONAL CALIBRATION

4.1. Basic Idea

Mathematically, the concept of 2-D LUTs for mapping input CMY to output C'M'Y' may be expressed as a cascade of two stages. In the first stage, two intermediate variables are computed from the input CMY control values for each output control value. In the second stage, 2-D LUTs are used to obtain the output C', M', and Y' control values as a function of the two corresponding intermediate variables. The 2-D LUT mapping is depicted in Fig. 5, and described mathematically below:

Stage 1: $(s_1, t_1) = v_{i1}(C, M, Y)$ 4(a)
$(s_2, t_2) = v_{i2}(C, M, Y)$ 4(b)
$(s_3, t_3) = v_{i3}(C, M, Y)$ 4(c)

Stage 2: $C' = f_1(s_1, t_1)$ 4(d)
$M' = f_2(s_2, t_2)$ 4(e)

$Y' = f_3(s_3, t_3)$ 4(f)

where s_k, t_k are intermediate variables that depend on the input CMY. Together, functions v_{ij} and f_i define the overall 2D calibration function $f_c(v_s)$ defined in Section 2.

4.2. Proposed Implementation

There are numerous choices for the functions v_{ij} that determine the two indices to the LUT. We adopted two main criteria for selecting these functions:

i) low computational complexity (since these functions are calculated for each image pixel);

ii) the functions should offer independent control along certain critical axes of interest, including individual channels, gray (C=M=Y), and secondary color mixtures.

We describe two different approaches for 1.) Printer calibration and 2.) CMYK Device emulation.

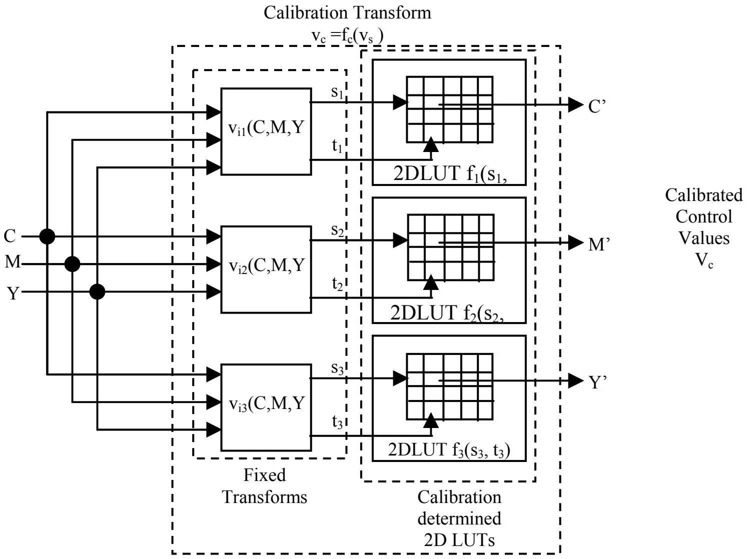


Figure 5. 2-D Calibration Transformation Architecture

4.2.1. Printer Calibration

For the calibration application, we define the intermediate variables for the LUTs as s_i, t_i as follows:

$s_1 = M + Y$; $t_1 = C$ 5(a)

$s_2 = C + Y$; $t_2 = M$ 5(b)

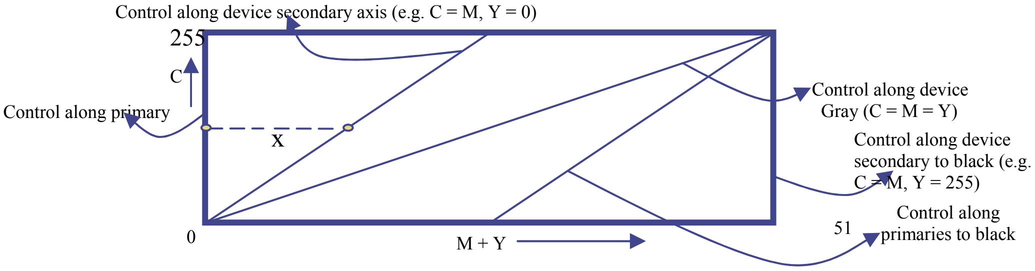
$s_3 = C + M$; $t_3 = Y$ 5(c)

In other words, the LUT f_1 in Fig. 5 for the cyan mapping is a function of two variables: i) input C and ii) sum of M and Y. Analogous relationships hold for f_2, f_3, the LUTs for Magenta and Yellow mapping.

Fig. 6 shows a conceptual representation of the 2-D domain of LUT f_1 (2-D LUT for Cyan), along with examples of certain critical axes along which the desired response characteristics can be maintained. The latter is done via the usual calibration process; namely printing, measuring (or modeling) the printer response along the chosen axes, and correcting to the desired response. The LUTs f_2 and f_3 are analogous. The greater control offered by a 2-D LUT when compared to 1D TRCs is now readily apparent. We can specify, for example, both a 1-D linearized response to ΔE from paper along the pure C axis, as well as a neutral 3D response in CIELAB along the C = M = Y axis (which maps to the line M + Y = 2C in Fig. 6). It is also possible with this scheme to simultaneously accomplish some level of control along device-secondary axes (i.e. along device-blue C = M, Y = 0; device green C = Y, M = 0; and device Red M = Y, C

= 0 axes respectively). On specifying the response along certain 1-D loci, the rest of the 2-D LUT may be filled in by interpolation. In our implementation we used linear interpolation along the (M+Y) direction.

Note also that for the 2-D calibration transforms in Fig. 5, the intermediate control variables (s_i, t_i) are only conceptually 2-D. For purposes of implementation, however, the 2-D LUT could be implemented by unraveling into a single 1-D table. For example the 2-D LUT f_1 could be replaced by a 1-D LUT addressed by (for example) C + 256 * (M + Y) for 8-bit input CMY values. For 8 bit CMY values the transformation C + 256 * (M + Y) maps different (C, M + Y) pairs to distinct values, so the proposed 1-D LUT version is only an equivalent of the 2-D version. Note also that a hybrid scheme may be adopted, where some device channels are handled by 2-D calibration transforms and others by traditional 1-D calibration transforms. Specifically, a one-dimensional TRC (e.g. ΔE from paper) may be used for the K channel in combination with the 2-D LUTs for C, M and Y as described above.


Control along device secondary axis (e.g. C = M, Y = 0)
255
C
Control along primary
X
0
M + Y
Control along device Gray (C = M = Y)
Control along device secondary to black (e.g. C = M, Y = 255)
51
Control along primaries to black

to blac

Figure 6. Conceptual Representation of the 2-D space of the Cyan LUT

4.2.2. CMYK Device emulation

In Section 2 we described calibration the process of "calibrating" or tuning a target device to emulate the color response of a reference device that may be an actual device, an industry standard or a mathematical idealization. While the three modes of specifying reference devices fall under the same conceptual umbrella, in practice the design of the calibration transform depends to a very large extent on the relationship between the reference device being emulated and the emulating device. We now describe a general procedure for the design of 2-D LUTs for CMYK device emulation.

The 2-dimensional emulation LUTs for C, M and Y are similar to those described before in Eqn. (5). In addition we choose a 2-D LUT f_4 for the K channel as well. For the K channel LUT, we have:

$(s_4, t_4) = v_{i4}(C, M, Y, K)$ 6(a)

$K' = f_4(s_4, t_4)$ 6(b)

$s_4 = \min(C, M, Y)$ 6(c)

$t_4 = K$ 6(d)

This is a reasonable choice as min(C, M, Y) is an approximate measure of the "black component" in the input. The technique for populating the LUTs for C, M, Y and K are however different in flavor from those used for calibration. We fill up these LUTs to mimic the response of the 4-dimensional (CMYK to CMYK) emulation transform. This is illustrated in Fig. 7 for the case of emulating a SWOP offset CMYK press on a Xerox digital CMYK printer. Note that, the mapping from the 4-dimensional space of emulated CMYK to the 2-D space of the emulation LUTs is not unique. The mapping is hence selected so that the 2-D transform agrees with the 4-D response in regions critical to color reproduction.

4.3. Performance Analysis

For 8 bit input CMY images, the proposed strategy requires three 2-D LUTs, each LUT is of size 256 x 511 = 128 Kbytes. Hence the overall size is just under 400 Kbytes, still likely an acceptable requirement in high-end systems. The computational processing comprises 3 additions to compute the s_i's and 3 2-D table look-ups per pixel. This is substantially cheaper than tetrahedral interpolation typically used in 3-D LUTs.[6]

As argued in the previous sub-section, the traditional trade-off between achieving linear responses along the individual separations or gray-balance (along the C = M =Y axis) is resolved by appropriately populating the corresponding axes in the 2-D table. More generally, note that any given input point (p_1, p_2) in the 2-D LUT of Fig. 5 corresponds to a line in CMY space formed by the intersection of the two planes C=p_1, M+Y = p_2. All points along this line will thus map to the same C'. This compromise is significantly less severe than the one encountered in 1D calibration, where an entire plane maps to the same output C'. The additional control may be used to overcome other device color problems. Often the hue along the printer's native blue axis (C = M) is purple and not really blue – and it may be desirable to use the calibration to produce a blue hue along the C = M axis. This can be achieved with the proposed architecture by appropriately populating the "white to secondaries" axis (M + Y = C axis in Fig. 6) in the Cyan and Magenta 2-D LUTs. (Note, several lines in the 3-D CMY space alias down to this axis including the C = M, Y = 0 axis) With the 2-D approach, this control on device blue does not compromise linearity along the individual separations or gray-balance. In general, responses may also be specified along arbitrary curves in the 2-dimensional space of the LUT (not necessarily lines, in Fig. 6) and the rest of the 2-D table can still be easily be filled via interpolation.

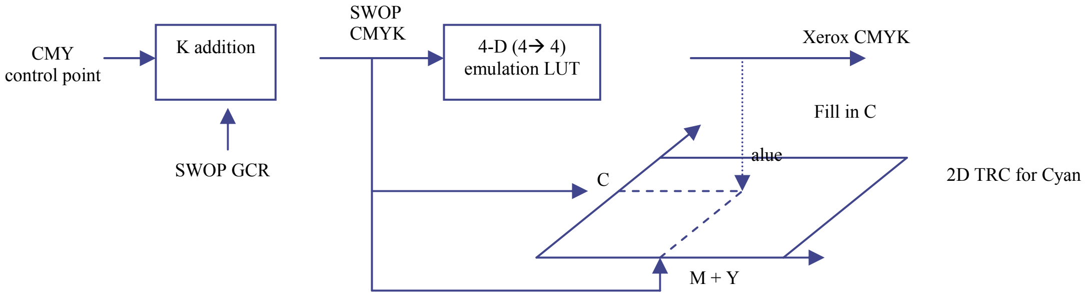


Figure 7. Derivation of 2-D emulation LUTs from 4-D corrections, Cyan 2-D LUT

As a final comment, 2-D calibration necessitates certain constraints on the preprocessing functions v_{i1}, v_{i2}, v_{i3}. Essentially, these functions must be monotonic with respect to input C, M, Y. Besides these, other constraints (e.g. smoothness) may be required in practice. These constraints however, are clearly less restrictive than those imposed by traditional one-dimensional calibration and also explain the greater flexibility offered by the 2-dimensional framework.

5. RESULTS

5.1. Device Calibration

An experiment was conducted with a Xerox Phaser 7000 CMYK laser printer. The 2D LUTs for C, M and Y were filled in as described in Section 4.2.1. A pure Cyan sweep with 16 levels of C = 0, 17, 34 … 255, M = 0, Y = 0 was processed via the two 1-D TRCs (*ΔE* from paper and gray-balance) and the 2-dimensional LUTs/TRCs. Subsequently, the processed sweep was printed and CIELAB values measured. The CIELAB difference from paper white was then computed and plotted (Fig. 8) against increasing value of digital count. By observing the plots we can see that a linear response is obtained for the 1-D *ΔE* from paper TRCs (as they were tuned for the same) as well as the 2-D corrections. In fact, the responses overlap (ignoring measurement noise). This is expected because the pure C, M and Y axes in the two-dimensional space spanned by the 2-D LUTs were filled with the respective one-dimensional channel independent corrections. The gray-balanced 1-D TRCs however do not give a linear response along the individual separations. Similar results can be observed along the Magenta and Yellow channels.

To evaluate gray-balance we process an equal CMY (C = M =Y) sweep via the three corrections, and print and measure the CIELAB values. Note, that a true neutral sweep would mean a* = b* = 0 in CIELAB co-ordinates. Hence, we define a metric similar to the ΔE_{ab} difference as

$$GB(d) = \sqrt{a^*(d)^2 + b^*(d)^2}\,,\ 0 \le d \le d_{\max} \tag{7}$$

where *a*(d)* and *b*(d)* are the *a** and *b** components of the CIELAB measurements of an equal CMY patch with digital value *d* (as before d = 0, 17, 34 … 255). The metric is hence defined to quantify the deviations from the true neutral. The average value of the metric for equally CMY sweeps processed via the three calibrations are tabulated in Table 1. The deviation is significantly lesser for the case of gray-balance TRCs as opposed to channel independent ΔE from paper. Hence, a linear response along the individual separations does not ensure gray-balance. The deviation for the gray-balanced TRCs and the 2-D corrections are identical. This is again an expected result of populating the C = M = Y axis in the 2-D C, M and Y LUTs with 1-D gray-balanced TRCs.

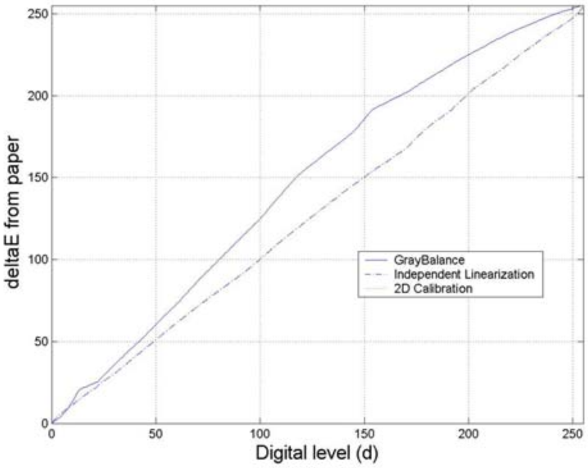


Figure 8. Channel-wise linearity vs gray-balance illustration for 1-D TRCs

Calibration Type	Average GB(d)
1-D channel independent	7.4055
1-D gray-balanced	1.1723
2-D	1.1723

Table 1. Deviations from true neutral for the three calibration methods

5.2. Stability control

The previous section investigated the ability to calibrate the device to a known state at a fixed instance in time. An equally important consideration is stability of this calibration over time. To examine this, the following experiment was performed. A calibration and characterization for the Phaser 7700 printer were derived at a given time T_0. The accuracy of the device correction-function was then evaluated by running a CIELAB target well within the printer gamut through the inverse characterization function to compute the CMYK needed to produce the device independent response. The subsequent calibration was applied to generate the C'M'Y'K' actually sent to the device. The target was then printed and CIELAB values measured. The CIELAB difference between the measured and actual specified CIELAB values provides a measure of the accuracy of the characterization/calibration. The same exercise as repeated for calibrations (1-D ΔE from paper, gray-balance and 2-D) derived at time $T_1 >> T_0$, while maintaining the same characterization derived at time $T_{0.}$ The stability experiment is illustrated in Fig. 9.

Table 2. shows the results of the experiment in the form of CIELAB errors computed between the "specified" and measured CIELAB values for the various calibrations derived at $T_{1.}$ In our experiment T_1 was twenty days after T_0. In addition, Table 2. also shows CIELAB errors for calibration derived and measured at T_0 and calibration derived at T_0 but measured at T_1. (Note that, all throughout the same characterization derived at T_0 was used.) The latter indicates the device drift over the period T_1 - T_0 and provides an upper bound on the errors while the former indicates the initial state of the device (i.e. how well was the printer characterized/calibrated to begin with) and hence forms a lower bound. The CIELAB errors for the new calibrations (at T_1) must hence be considered relative to these bounds. For the new calibrations (1-D gray-balance, *ΔE* from paper and 2-D) derived and measured at T_1, the lowest errors were recorded for 2-D calibration. Intuitively, this is expected because the 2-D calibration process pins-down the desired response at more locations.

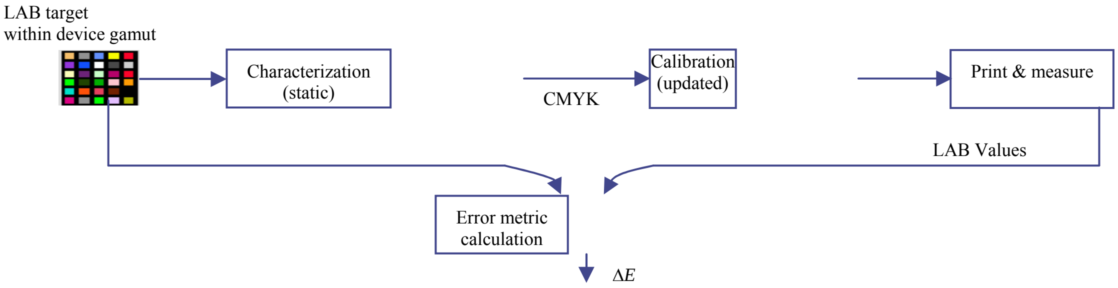


Figure 9. Stability experiment for evaluating 1-D Vs 2-D calibrations

Correction	Derived at	Measured at	Average ΔE_{94} error	95% ΔE_{94} error
1-D graybalance + characterization	T_0	T_0	2.21	4.08
1-D channel independent	T_1	T_1	5.78	7.51
1-D graybalance	T_1	T_1	4.73	8.02
2-D	T_1	T_1	2.66	4.59
No recalibration	T_0	T_1	6.83	10.67

Table 2. Stability: CIELAB errors measured for various calibrations

5.3. CMYK Device emulation

We now compare 1-D vs 2-D transforms for device emulation. We consider the emulation of the industry standard SWOP offset CMYK[16] on a Xerox laser CMYK printer.

The 1-D emulation transforms are derived as channel independent TRCs from SWOP CMYK to Xerox CMYK, where in each channel is derived to mimic the channelwise response (ΔE from paper) of SWOP on the Xerox device. The 2-D emulation LUTs for C, M, Y and K are derived from the 4-D transform as described in Section 4.2.2. The 1-D and 2-D techniques are then compared against the 4-D emulation transform, the latter being treated as ground truth. We generate a set of SWOP CMYK values uniformly distributed over the SWOP gamut. These values are generated from an N x N x N (N = 32) grid of CMY values processed through a typical SWOP GCR strategy. The 1-D, 2-D and 4-D emulation transforms are then used convert from SWOP CMYK to Xerox CMYK. A Neugebauer printer model for the Xerox digital printer[11] is used to predict the device independent CIELAB values for the three cases. The CIELAB difference between 1-D (or 2-D) and the 4-D LAB values gives a measure of deviation from the ideal (4-D) emulated response. Table 3. shows the average, 95% and maximum CIE94 errors for the 1-D and 2-D cases. Clearly, the errors are lower for 2-D emulation. In particular, the 95% point and maximum errors for 2-D are much lower than 1-D emulation.

Emulation Type	Average ΔE_{94} error	95 % ΔE_{94} error	Max. ΔE_{94} error
1-D emulation	4.7825	14.5568	16.5688
2-D emulation	2.2783	9.83	12.374

Table 3. Quantitative evaluation of 2-D Vs 1-D emulation using printer model.

6. CONCLUSIONS

A novel two-dimensional calibration technique for output devices is described. This technique enables significantly greater control and stability over regions of the color space that cannot be controlled simultaneously by 1-D calibration transforms. The 2-D scheme can also be used for effectively emulating one device with another. In terms of calibration, stability control, and device emulation, the 2-D scheme offers far superior quality compared to the 1-D equivalent, while incurring a relatively modest computational overhead that is likely to be acceptable for real-time processing of color images. While the ideas and results have been presented for printers, the concepts can be readily applied to additive color devices, e.g. CRT displays or LCDs.

7. REFERENCES

1. R Bala, *Device Characterization - Digital Color Imaging Handbook, Chapter 5*. CRC Press, 2003.
2. E. J. Giorgianni, T. E. Madden and K. E. Splauding, *Color Management for digital imaging systems - Digital Color Imaging Handbook, Chapter 4*. CRC Press, 2003
3. G. Sharma, *Color Fundamentals for Digital Imaging - Digital Color Imaging Handbook, Chapter1*. CRC Press, 2003
4. R. W. G. Hunt, *Measuring Colour.* Prentice Hall, 1991
5. "International Color Consortium", www.color.org
6. R. Bala and R.V. Klassen, *Efficient color transformation implementation – Digital Color Imaging Handbook, Chapter 11.* CRC Press, 2003
7. J.Z. Chan, J.P. Allebach and C.A. Bouman, "Sequential Linear Interpolation of Multidimensional Functions," *IEEE Transactions on Image Processing*, vol. 6, 1231-1245, Sept. 1997
8. H.R. Kang, *Color Technology for Electronic Imaging Devices*, SPIE, Bellingham, WA, 1997
9. P.C.Hung, "Colorimetric calibration in electronic imaging devices using a look-up table model and interpolations", *Journ. Electronic Imaging*, vol 2, no. 1, pp. 53-61, 1993
10. GSharma, RBala, MMaltz, LKMestha, J.P. Van de Capelle, "Two-Dimensional Cailbration architectures," US patent filed May '03.
11. R. Bala, "Optimization of the Spectral Neugebauer model for printer characterization", *J. Electronic Imaging,* 8(2), 156-166, 1999
12. M. Xia, E. Saber, G. Sharma, and A. M. Tekalp, ``End-to-end color printer calibration by total least squares regression ," *IEEE Trans. Image Proc.*, vol. 8, no. 5, pp. 700-716, May 1999.
13. A. Artusi, and A. Wilkie, "Novel color printer characterization model," *J. Electronic Imaging*, vol. 12, 448-459, July 2003
14. R. Balasubramanian, "Refinement of printer transformations using weighted regression", *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts,* vol. 2658, 1996
15. R. Balasubramanian, S. Wang, "Color balanced TRC correction to compensate for illuminant changes in printer characterization", *US patent DA0441,* filed May 15, 2000
16. "Specifications Web Offset Publications", www.swop.org
17. "What is sRGB – Introduction to the standard default RGB color space developed by Hewlett-Packard and Microsoft," http://www.srgb.com/aboutsrgb.html