Two-Dimensional Transforms for Device Color Correction and Calibration

Raja Bala, Gaurav Sharma, *Senior Member, IEEE*, Vishal Monga, *Member, IEEE*, and Jean-Pierre Van de Capelle

***Abstract*—Color device calibration is traditionally performed using one-dimensional (1-D) per-channel tone-response corrections (TRCs). While 1-D 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 per separation (or per-channel), TRCs in a printer can be used to either ensure gray balance along the $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ axis or to provide a linear response in delta-E units along each of the individual (C, M, and Y) axis, but not both. This paper proposes a novel two-dimensional color correction architecture that enables much greater 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. Superior cost quality tradeoffs (over 1-D methods) are also achieved for emulation of one color device on another.**

***Index Terms*—Color calibration, color control, device characterization, display, printer calibration, printing.**

I. 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], [3]. 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 [4] are popular choices for the device independent representation [5]–[7]. Fig. 1 illustrates color management via device characterization and calibration. For succinctness, we assume familiarity with basic color science terms and concepts throughout this paper. Readers unfamiliar with the terminology are referred to tutorial/review articles [8], [9] that present these concepts in a signal processing context and to [10] for comprehensive coverage.

This paper focuses on the calibration function [11]–[13]. 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]. For example, for a three-color (CMY) printer, the goal of calibration is to determine a transform from CMY (device values as determined by the characterization) to $\mathrm{C'M'Y'}$ (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 (1-D) TRC correction to the individual channels of the device. Mathematically, the TRC mapping can be defined as $\mathrm{C'} = \mathrm{f_1(C)}, \mathrm{M'} = \mathrm{f_2(M)}, \mathrm{Y'} = \mathrm{f_3(Y)}$. A significant benefit in using 1-D TRCs is that these are very efficient for real-time processing of color images. This process assumes that by applying this calibration the printer response will be essentially stabilized throughout the entire CMY color space.

Examples[1] of responses specified for 1-D calibration include 1) linearization of each channel to delta-E (ΔE) from paper and 2) linearization and gray-balance so that $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ or some other combination of CMY, maps to the L^* axis in CIELAB space [1]. 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 $\mathrm{M} - \mathrm{Y}$ plane perpendicular to the C axis) will map to the same $\mathrm{C'}$ 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 $\mathrm{C} = \mathrm{M}$, and $\mathrm{C} = \mathrm{K} = 0$). Conversely, if the TRCs are independently linearized, this will generally not yield gray balance for $\mathrm{C} = \mathrm{M} = \mathrm{Y}$.

As an alternative to control of each individual channel for calibration, the calibration may be performed as a full three-dimensional (3-D) function from input $\mathrm{C}, \mathrm{M}, \mathrm{Y}$ to output $\mathrm{C'}, \mathrm{M'}, \mathrm{Y'}$. Mathematically stated, $\mathrm{C'} = \mathrm{f_1(C, M, Y)}, \mathrm{M'} = \mathrm{f_2(C, M, Y)}, \mathrm{Y'} = \mathrm{f_3(C, M, Y)}$. With 3-D corrections, the

Manuscript received April 5, 2004; revised July 26, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gabriel Marcu.

R. Bala and J.-P. Van de Capelle are with the Xerox Corporation, Webster, NY 14580 USA (e-mail: rbala@crt.xerox.com; Jean-pierre.van.de.Capelle@usa.xerox.com).

G. Sharma is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: gsharma@ece.rochester.edu).

V. Monga is with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: vishalmongahere@yahoo.co.in).

Digital Object Identifier 10.1109/TIP.2005.851678

[1]These calibration techniques are described in greater detail in Section IV, where the associated terms are also defined.

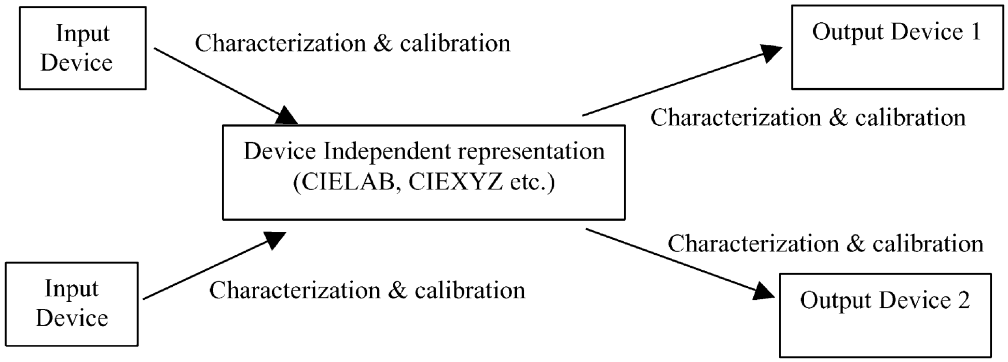


Fig. 1. Device-independent color management via characterization and calibration.

desired response can be specified at any point in 3-D 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-D lookup tables (LUTs). A full resolution 3-D LUT with direct lookup is usually considered prohibitively large. For 8-bit processing for instance, a full 3-D LUT would require $3^*(256)^3$ bytes $=$ 48 MB of storage. Alternatively, sparse 3-D LUTs can be used in conjunction with some form of 3-D interpolation [14]–[18]. In this case, the interpolation operation can be too computationally intensive for high-speed printing applications. Also, the 3-D interpolation may introduce undesirable image quality defects. 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 four-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 [19]. The 2-D LUTs offer a very attractive compromise between 1-D and 3-D calibration both in terms of quality and complexity. When compared with 1-D calibration, the additional dimension afforded by a 2-D LUT offers a greater degree of control on the device response; improved temporal stability; and an improved ability to emulate the response of a different device. At the same time, the memory and computational complexity 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 number of measurements required for deriving the 2-D LUTs is also of the same order as traditional 1-D methods.

Section II briefly presents background material on device calibration, characterization and emulation. Section III describes a general framework for device color correction that encompasses both calibration and emulation. Section IV presents traditional 1-D printer calibration methods and their limitations. The proposed 2-D transform is presented in Section V, along with applications to calibration, stability control, and emulation. Results are presented in Section VI, and concluding remarks are collected in Section VII.

II. 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 [20]–[23], or empirical [1], or a hybrid between the two [24].
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. 2 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., linearity, gray balance, smoothness, monotonic response, etc., prior to derivation of the characterization. Second, temporal variations in the device's response, and variations within a fleet of similar devices can often be addressed in the calibration function. Third, other settings that impact the device color response, for example, the choice of the halftone scheme, media, or viewing illuminant [25] may be compensated for to some extent in the calibration process.

Finally, there is another important benefit to the partitioning in Fig. 2. Recall that calibration involves specifying and main-

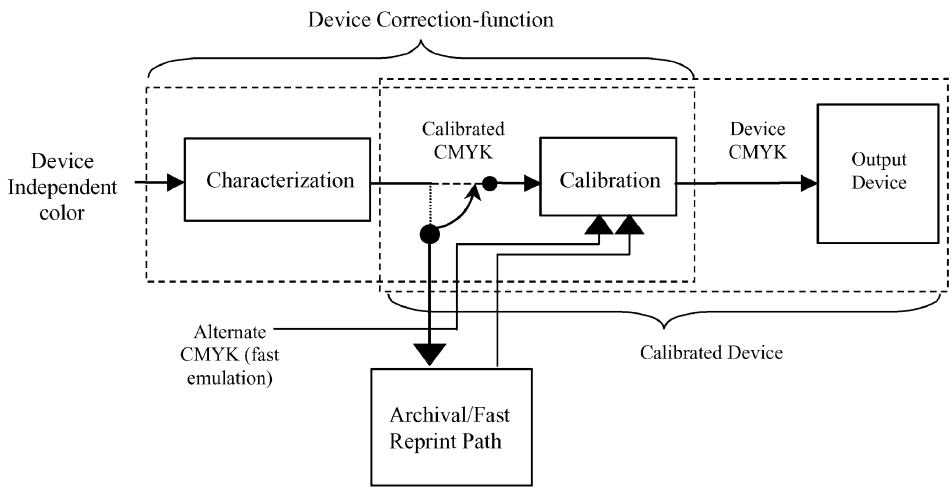


Fig. 2. Schematic representation of the partitioning of the device-correction function into characterization and calibration.

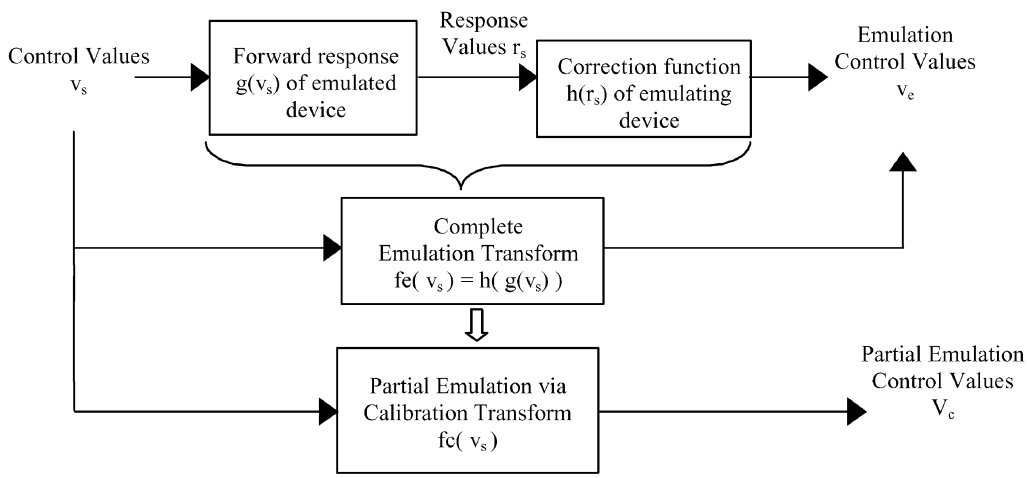


Fig. 3. Emulation of one device on another through concatenation of emulated device's forward response and emulating device's device correction function.

taining 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 1) an actual device, 2) an industry standard device (e.g., SWOP offset press [26] or sRGB display [27]), or 3) 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.

Device emulation involves mimicking the color response of one device (the emulated device) with another (the emulating device). The goal in device emulation is, thus, 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. 3. In general, the emulation transform is a computation and memory intensive multidimensional function. For example, if the emulated and emulating devices are both CMYK printers, the emulation transform is a four-dimensional (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.

From the aforementioned discussion, it should be clear that the purpose of the calibration transformation is to facilitate a tradeoff. Unlike the full device-correction function, the calibration transformation provides control of the output device only in a limited fashion. However, in comparison to the full device-correction function the calibration transformation 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.

Among the various objectives that calibration can accomplish, in this paper we particularly explore and evaluate three aspects of the novel calibration architecture that we develop: 1) the extent of control in various regions of color space, 2) device emulation accuracy, and 3) temporal stability.

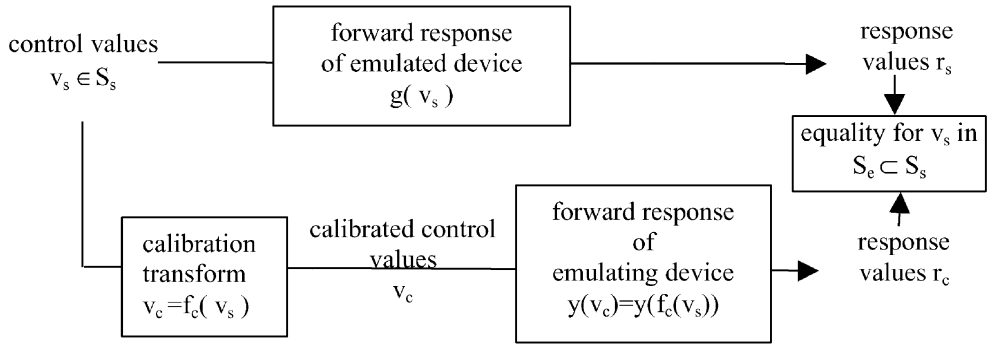


Fig. 4. General diagram of partial emulation through calibration.

III. A General Framework for Color Correction

In this section, a common framework for device color correction is presented that unifies the concepts of device calibration and device emulation. Essentially, the problem of calibrating a device to achieve specific color characteristics is more generally restated as the problem of emulating a reference device that possesses those characteristics. In our mathematical notation, we use subscript "e" for variables related to an *emulating* device and the subscript "s" for variables asscociated with an emulated device (often a *standard* or reference device). Likewise, for functions the subscript "e" is utilized for a general emulation transform and the subscript "c" for a (more constrained) calibration transform. With these conventions, we define the following symbols.

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., CMYKcube)
3) $g(\,)$—forward device response function for the emulated device (e.g., CMY $\rightarrow$ CIELAB mapping).
4) $g(\mathbf{v}_s)$—the emulated device's response to the control values $\mathbf{v}_s$ (measured for instance in CIELAB).
5) $\mathbf{v}_e$—a vector of control values in the device space of the emulating device (e.g., CMY/CMYK).
6) $y(\,)$—forward device response function for the emulating device (e.g., CMY $\rightarrow$ CIELAB mapping).
7) $y(\mathbf{v}_s)$—the emulating device's response to the control values $\mathbf{v}_s$ (measured for instance in CIELAB).
8) $\mathbf{r}_s$—a vector of response values (e.g., CIELAB, or ΔE from paper).
9) $h(\,)$—the emulating device's correction function (e.g., mapping from CIELAB to printer CMY/CMYK).
10) $h(\mathbf{r}_s)$—one vector of control values in the control space of the emulating device that produces the response $\mathbf{r}_s$.
11) $f_e(\,)$—full emulation transform (emulated device's values $\rightarrow$ emulating device's control values).
12) $f_c(\,)$—calibration transformation.
13) S_e—the region in device space wherein calibration coincides with full emulation. This can be described mathematically as

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

The calibration process may be seen as a computationally restricted version that provides agreement with the multidimensional emulation transform within the limited set S_e, as described in (1). While this expression defines the emulation region in the domain of the input control values, the emulation region may also be defined in terms of the emulated device's response domain (for example CIELAB), where the corresponding region is given by $R_e = g(S_e)$

$$y(f_c(\mathbf{v}_s)) = g(\mathbf{v}_s) \quad \forall \mathbf{v}_s \in S_e. \tag{2}$$

Within the emulation region S_e, the response of the emulated device to a set of control values $\mathbf{v}_s$ is identical to the response of the emulating device to the signals obtained by mapping the control values $\mathbf{v}_s$ through the calibration transformation $f_c(\,)$. This is illustrated in Fig. 4. This is a more general description of emulation than the one represented in Fig. 3. Typical device correction functions map 3-D device independent color specifications to device control values that produce these (device independent) response values, and the emulation shown in Fig. 3 implicitly assumes this. The response values being emulated, $\mathbf{r}_c$ and $\mathbf{r}_s$ in Fig. 4, however, need not correspond necessarily to colorimetry alone (e.g., they could be densitometric values). In general they need not necessarily be 3-D (e.g., they may be simply the color difference ΔE evaluated from white paper—a commonly used metric for linearization of individual TRC's). Alternatively, these signals may be of higher dimensionality (e.g., spectral or colorimetric under multiple illuminants). Finally, the responses may be of different dimensionality for different points in the emulation set S_e. An example of this is the control of a 1-D response along individual C, M, Y axes, and 3-D gray-balanced response along the device gray $(C = M = Y)$ axis.

Note that the emulation cannot encompass a region outside the gamut of the emulating or emulated device. For the case of partial emulation this means that $R_e = g(S_e)$ is a subset of the intersection of emulated and emulating device's gamut. For the case, when the response is specified in CIELAB, the requirement is that R_e be a subset of the common (intersection) CIELAB gamut of both printers. In cases, where the same response values may be produced by different choices of the emulating device control values additional constraints may be imposed on the emulation transform. For instance, for emulation using a CMYK printer, it is desirable that the emulation not only results in a color match (under specified illuminant) but also that the CMYK values obtained from the calibration conform to those obtained from the normal Gray component replacement (GCR) strategy used for the emulating device [1].

It is worth re-emphasizing here that our definition of emulation encompasses the case of both existing physical devices and virtual devices such as those defined by standards or ideal mathematical constructs. In fact, in the calibration case, the emulated device is indeed an ideal mathematical concept. For the purposes of calibration, it is not necessary to know the complete response of the device to be emulated; it is sufficient if the response is known within the region S_e in which the emulating device needs to match the emulated device through calibration.

IV. One-Dimensional Calibration

Calibration traditionally involves 1-D 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 three-color CMY calibration. The calibration architecture for a CMY printer is shown in Fig. 5. The two most common examples of 1-D calibration are described next, namely linearized ΔE response along the individual channels, and gray-balanced response.

A. Channelwise Linearization to ΔE From Paper

This technique is motivated by a desire to calibrate each device channel independently such that the resulting response is linear in perceptual units. Since color perception is 3-D, whereas each channel offers only a single control, a simplification is necessary. Commonly, the overall perceptual response is quantified by the ΔE color difference with respect to the printing medium ("paper white"), where ΔE (more formally denoted ΔE^*_{ab}) is the Euclidean distance in the (approximately) perceptually uniform CIELAB color space [5]. For this type of calibration, each channel is independently linearized to the ΔE color difference between the ith ($i = C, M, Y$, 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 in ΔE from paper. This response may be defined as[2]

$$\begin{aligned} r_s = g(v_s) &= \Delta E_{ab}(c_{\text{medium}}, c(v_s)) \\ &= \frac{v_s}{v_s^{\max}} \left\{ \Delta E_{ab} \left(c_{\text{medium}}, c\left(v_s^{\max}\right)\right)\right\} \end{aligned} \tag{3}$$

where

v_s input control value for one of C, M, and Y channels;
$v_s^{\max}$ maximum input control value (e.g., 255 for 8-bit inputs);

[2]For notational simplicity, we assume that each channel offers a monotonic response in ΔE from paper. Situations which deviate from this assumption can be handled by suitable modifications.

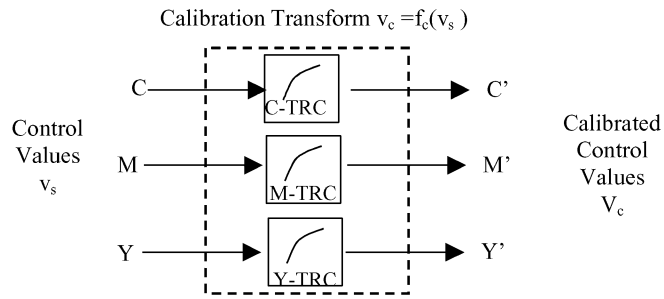


Fig. 5. Traditional 3-color 1-D calibration transformation.

c_{medium} CIELAB measurement of the medium;
$c(v_s)$ CIELAB response corresponding to input value v_s.

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

B. Gray-Balanced Calibration

Due to the greater perceptual significance of neutral or achromatic colors [28], a more accurate control of these colors is often desirable in color reproduction systems. For this reason, it is sometimes preferable to use device calibration in order to control the device response to a known state along the L^* or lightness axis in CIELAB, an approach referred to as *gray balanced calibration*. Recall that the L^* axis corresponds to the line $a^* = b^* = 0$, i.e., the 1-D line in 3-D CIELAB space over which the color is perceptually achromatic or gray, varying between white and black along the two extremes. For three-color gray balancing, the response in CIELAB is specified along the device gray/neutral axis (defined as a curve in the CMY space) as the combined requirement that CMY values along this curve map to the L^* axis in CIELAB and that equal steps along this CMY curve correspond to equal steps in L^* (this is readily generalized to an arbitrary response curve in L^*.) The device gray axis is a suitably chosen curve in device color space, over which an achromatic response is desired. In the remainder of this paper the device neutral axis is chosen to be $C = M = Y$; generalization to alternate curves is trivial. 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 emulating device's C, M, Y that produce the desired neutral color. Many search techniques exist that can suitably address this problem [1].

There are two major motivations for gray-balance calibration. First, as mentioned earlier, the human visual system is particularly sensitive to color differences near neutrals. Second, 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, when it is opted to gray balance a printer along $C = M = Y$, most printers cannot be gray- balanced beyond a level $v_s^{\text{maxgray}} < v_s^{\max}$ where the TRC output v_c corresponding to one of the C, M, Y channels saturates 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

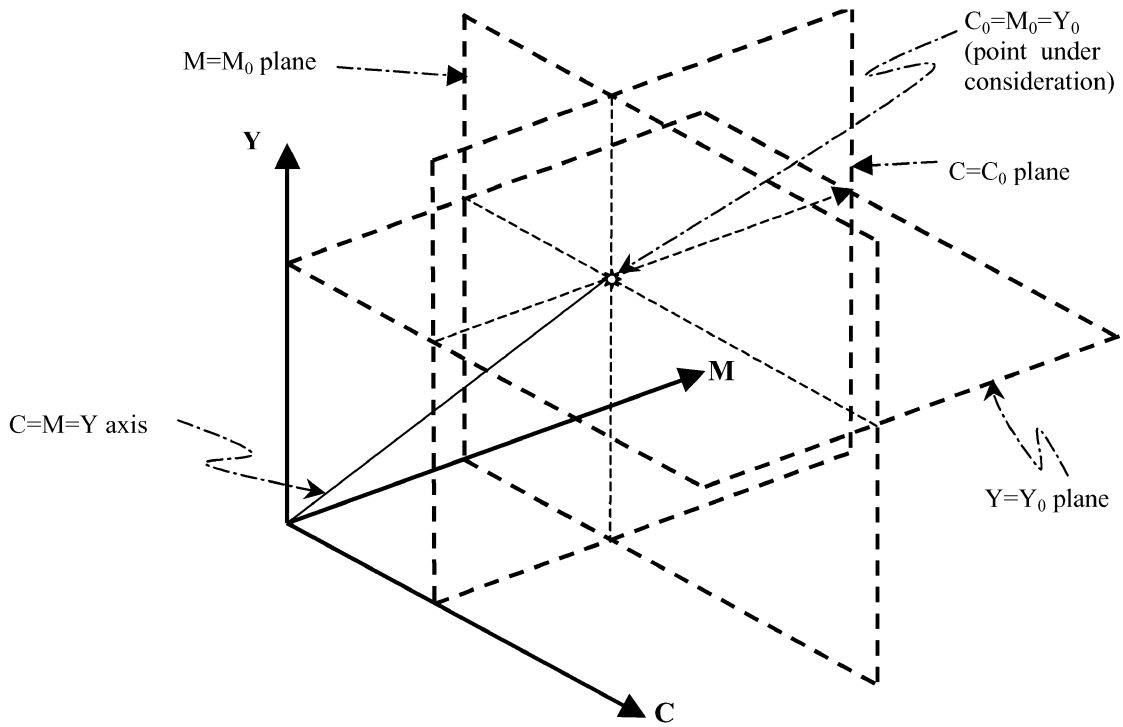


Fig. 6. Illustration of influence of calibration controls outside the emulation set for traditional 1-D TRCs.

an 8-bit system). Furthermore, this compromise must be arrived at smoothly in order to avoid discontinuities near the printer black point. A detailed description of this process can be found in [1].

C. 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. For example, 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.

In terms of output quality, however, 1-D TRCs severely limit the amount of control that can be exercised. Referring to the emulation framework introduced in Section III, the region of control $\mathrm{S_e}$ over which a precise device response can be achieved via 1-D TRCs comprises at most a set of 1-D loci in CMY space. In the case of channelwise linearization, $\mathrm{S_e}$ is made up of the three primary $\mathrm{C}, \mathrm{M}, \mathrm{Y}$ axes. As one moves away from these axes, the printer response becomes an uncontrolled outcome of the calibration process. In particular, since the $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ axis does not lie within $\mathrm{S_e}$, a channelwise linearization cannot ensure a gray-balanced response along this axis. Conversely, in the case of gray-balance calibration, the region of control $\mathrm{S_e}$ is the $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ axis. Now, the primary $\mathrm{C}, \mathrm{M}, \mathrm{Y}$ axes lie outside the region of control, and the calibration may result in an undesirable nonlinear device response along these axes. Specific quantitative examples of these effects will be shown in the results section.

Let us visit the aforementioned concepts from a geometric perspective. Assuming the case of 1-D gray-balance calibration, it is readily seen that the output value from the cyan TRC influences not only the corresponding point along the $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ axis, but also all other points in input CMY space having the same cyan value. This idea is illustrated in Fig. 6. The specified response to be emulated at the point $\mathrm{C_0} = \mathrm{M_0} = \mathrm{Y_0}$ determines the corresponding output value for the cyan 1-D TRC. However, this output value also influences the entire $\mathrm{C} = \mathrm{C_0}$ plane, and may, in fact, produce an undesirable response for colors on this plane other than (especially those distant from) the $\mathrm{C_0} = \mathrm{M_0} = \mathrm{Y_0}$ point. Analogous observations hold for the magenta and yellow TRCs.

Note that this coupling of the response in regions outside the control regime $\mathrm{S_e}$ necessitates a constraint that the 1-D TRCs are monotonically increasing functions along each of the three dimensions $\mathrm{C}, \mathrm{M}, \mathrm{Y}$. This constraint may itself limit the degree of control that can be exercised in certain regions of CMY space.

V. Two-Dimensional Calibration

A. Basic Principle of 2-D Calibration

Mathematically, the concept of 2-D LUTs for mapping input CMY to output $\mathrm{C'M'Y'}$ 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 $\mathrm{C'}, \mathrm{M'}$, and $\mathrm{Y'}$ control values as a function of the two corresponding intermediate variables. The 2-D LUT mapping is depicted in Fig. 7, and described mathematically as follows:

$$\text{Stage 1: } (\mathrm{s_1}, \mathrm{t_1}) = \mathrm{v_{i1}}(\mathrm{C}, \mathrm{M}, \mathrm{Y}) \tag{4a}$$

$$(\mathrm{s_2}, \mathrm{t_2}) = \mathrm{v_{i2}}(\mathrm{C}, \mathrm{M}, \mathrm{Y}) \tag{4b}$$

$$(\mathrm{s_3}, \mathrm{t_3}) = \mathrm{v_{i3}}(\mathrm{C}, \mathrm{M}, \mathrm{Y}) \tag{4c}$$

$$\text{Stage 2: } \mathrm{C'} = \mathrm{f_1}(\mathrm{s_1}, \mathrm{t_1}) \tag{4d}$$

$$\mathrm{M'} = \mathrm{f_2}(\mathrm{s_2}, \mathrm{t_2}) \tag{4e}$$

$$\mathrm{Y'} = \mathrm{f_3}(\mathrm{s_3}, \mathrm{t_3}) \tag{4f}$$

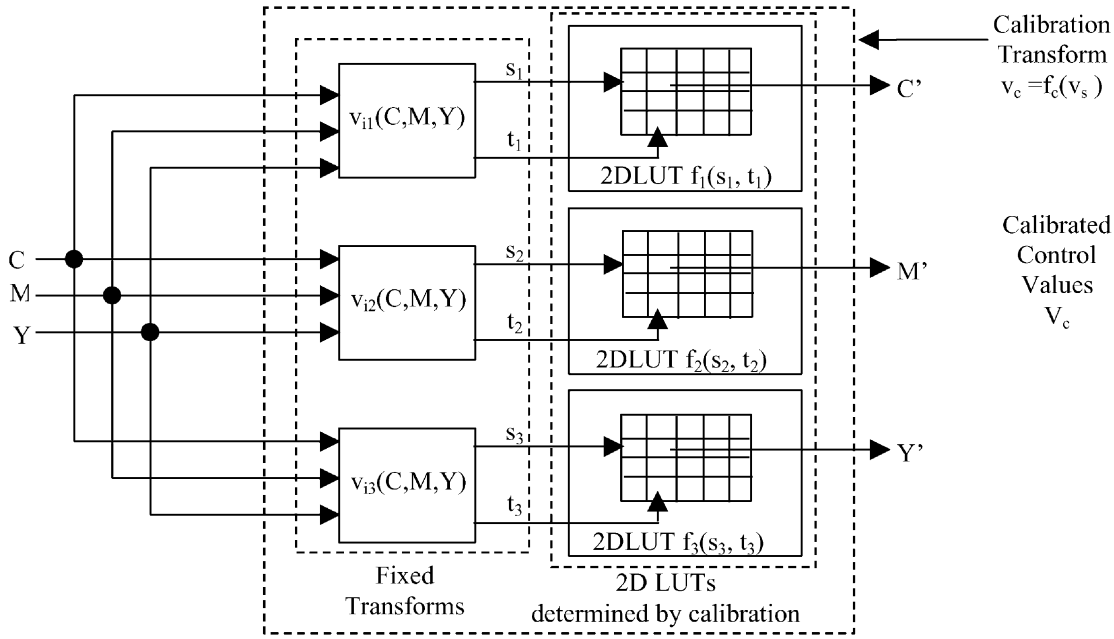


Fig. 7. Two-Dimensional Calibration Transformation Architecture.

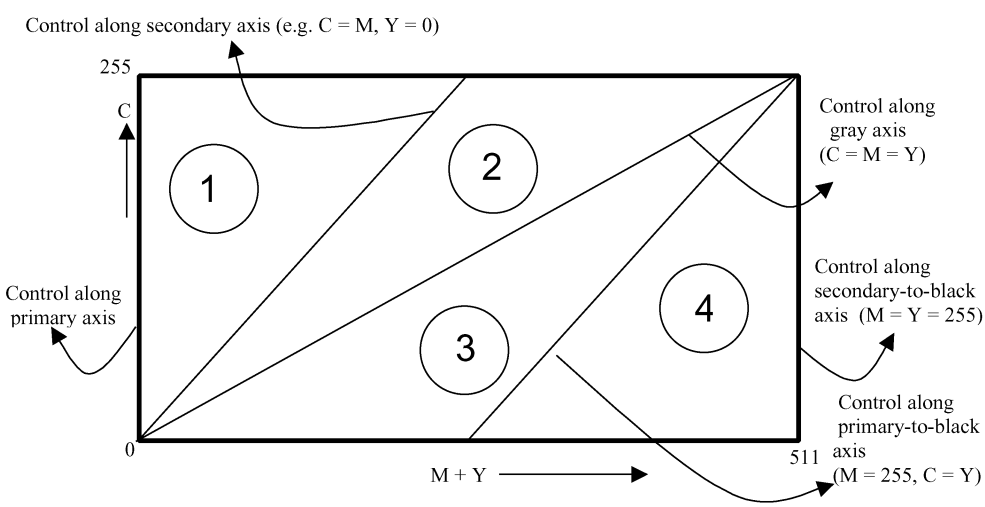


Fig. 8. Conceptual Representation of the 2-D space of the Cyan LUT. Various axes of control are included. These divide the 2-D space into four regions labeled 1–4.

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

B. 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:

1) low computational complexity (since these functions are calculated for each image pixel);
2) 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 our approaches for the two applications: 1) printer calibration and 2) CMYK device emulation.

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

$$s_1 = M + Y; \quad t_1 = C \tag{5a}$$
$$s_2 = C + Y; \quad t_2 = M \tag{5b}$$
$$s_3 = C + M; \quad t_3 = Y. \tag{5c}$$

In other words, the LUT f_1 in Fig. 7 for the cyan mapping is a function of two variables: 1) input C and 2) sum of M and Y. Analogous relationships hold for the LUTs for Magenta and Yellow mapping f_2, f_3.

Fig. 8 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 character-

istics 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 greater control offered by a 2-D LUT when compared to 1-D 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 3-D response in CIELAB along the $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ axis (which maps to the line $\mathrm{M}+\mathrm{Y} = 2\mathrm{C}$ in Fig. 8). It is also possible with this scheme to simultaneously accomplish some level of control along device-secondary axes (i.e., along device-blue $\mathrm{C} = \mathrm{M}, \mathrm{Y} = 0$; device-green $\mathrm{C} = \mathrm{Y}, \mathrm{M} = 0$; and device-red $\mathrm{M} = \mathrm{Y}, \mathrm{C} = 0$ axes, respectively). On specifying the response along certain 1-D loci, the rest of the 2-D LUT may be filled in by interpolation. For our implementation, we used linear interpolation along the $(\mathrm{M}+\mathrm{Y})$ direction.

Note, also, that, for the 2-D calibration transforms in Fig. 7, the intermediate control variables $(\mathrm{s_i}, \mathrm{t_i})$ are only conceptually 2-D. For purposes of implementation, however, the 2-D LUT could be implemented as a 1-D table by unraveling the 2-D LUT into a single 1-D table. For example the 2-D LUT $\mathrm{f_1}$ could be replaced by a 1-D LUT addressed by (for example) $\mathrm{C}+256^*(\mathrm{M}+\mathrm{Y})$ for 8-bit input CMY values. For 8-bit CMY values the transformation $\mathrm{C} + 256^*(\mathrm{M} + \mathrm{Y})$ maps different $(\mathrm{C}, \mathrm{M} + \mathrm{Y})$ 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 1-D 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.

2) CMYK *Device Emulation:* In Section III we presented a general framework for calibration as one where a target device is "calibrated" or tuned 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 intermediate variables for the 2-D emulation LUTs for C, M, and Y are similar to those described in (5). In addition, to afford greater control of the K channel, we introduce a fourth 2-D LUT $\mathrm{f_4}$ for this channel as well. For the K channel LUT, we have

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

$$\mathrm{K}' = \mathrm{f_4}(\mathrm{s_4}, \mathrm{t_4}) \tag{6b}$$

$$\mathrm{s_4} = \min(\mathrm{C}, \mathrm{M}, \mathrm{Y}) \tag{6c}$$

$$\mathrm{t_4} = \mathrm{K}. \tag{6d}$$

The technique for populating the LUTs for C, M, Y, and K is, however, different in flavor from those used for calibration. We fill these LUTs to mimic the response of the 4-D (CMYK to CMYK) emulation transform. This is illustrated in Fig. 10. We begin by describing one exemplary method of populating the

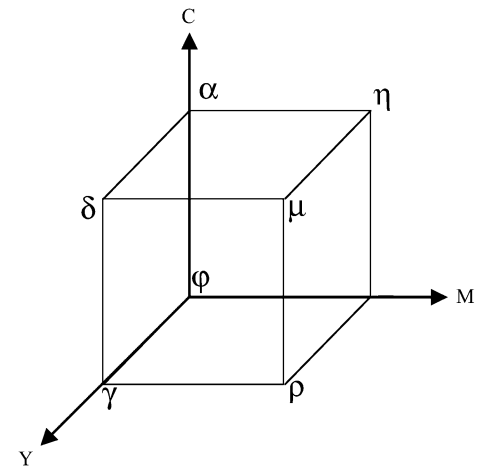


Fig. 9. Three-dimensional color cube. Triplets of vertices signify triangles corresponding to the four regions in Fig. 8.

K channel. The first row of the K channel LUT, i.e., $\mathrm{K} = 0$, is populated as follows: We run CMYK sweeps with $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ from $0, 1, 2 \ldots 255$ and $\mathrm{K} = 0$ via the 4-D emulation transform and fill in the resulting $\mathrm{C}', \mathrm{M}', \mathrm{Y}', \mathrm{K}'$ into the 2-D LUT at the cell location: $\mathrm{s_4} = \min(\mathrm{C}, \mathrm{M}, \mathrm{Y}), \mathrm{t_4} = \mathrm{K}$. The other 255 rows in the K LUT are filled in similarly. Again, several curves in the 4-D space alias down to the min(C, M, Y) axis in the 2-D space spanned by the K channel LUT. Our approach is based on choosing the response along the $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ axis. This is a reasonable choice as min(C, M, Y) is an approximate measure of the "gray component" in the input.

Next we discuss an exemplary method of populating the cyan 2-D LUT. Analogous arguments hold for magenta and yellow LUTs. The 2-D space in Fig. 8 may be divided into four regions (labeled 1–4 in the figure).

- Region 1 is the region bounded by the white-to-cyan $(\mathrm{M}+\mathrm{Y} = 0$ or $\mathrm{M} = 0, \mathrm{Y} = 0)$ and the white to secondaries $(\mathrm{M} + \mathrm{Y} = \mathrm{C})$ axes.
- Region 2 is the region bounded by the white-to-secondaries and white to black $(\mathrm{M} + \mathrm{Y} = 2\mathrm{C})$ axes.
- Region 3 is region bounded by the white-to-black and primaries to black $(\mathrm{M} + \mathrm{Y} = \mathrm{C} + 255)$ axes.
- Region 4 is the region between the primaries-to-black and secondary to black $(\mathrm{M} + \mathrm{Y} = 510)$ axes.

Fig. 9 shows a 3-D CMY color cube that we will use to illustrate the method used to populate the cyan 2-D LUT. In general, since Fig. 8 has one dimension less than Fig. 9, it is not possible to uniquely associate the aforementioned regions with unique regions in Fig. 9. Each of the four regions in the Cyan 2-D LUT is populated by a two-step process.

1) Select a set of one or more planar triangles in Fig. 9 (specified by a triplet of vertices) that all map to the (same) chosen triangular region in the 2-D LUT.
2) For each cell in the chosen triangular region of the 2-D LUT, determine the CMY points on the planar triangles selected in the first step that map to the cell coordinates. For each CMY point, select an amount of K to be added that is appropriate for the emulated device. Process each of the resulting CMYK points through the desired 4-D

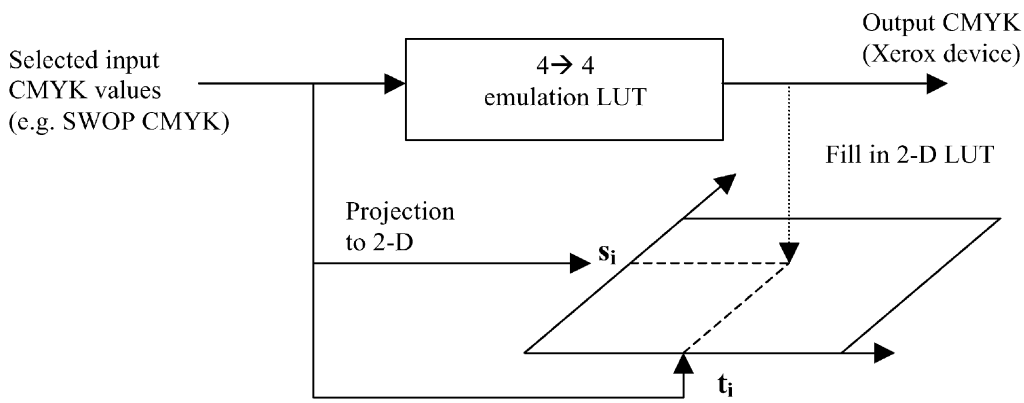


Fig. 10. Derivation of 2-D emulation LUTs from 4-D corrections, Cyan 2-D LUT.

emulation transform to obtain output $C'M'Y'K'$ values. The average of the cyan values C' is then stored at the chosen cell location in the 2-D LUT.

The triangles (see Fig. 9) associated with each region are:

Region 1: Triangles (φ, α, η) and $(\varphi, \alpha, \delta)$. Note that this approach is based on mapping the point $C = 255, M + Y = 255$ in 3-D CMY space to secondaries Green $(C = 255, M = 0, Y = 255)$ or Blue $(C = 255, M = 255, Y = 0)$.

Region 2: Triangles (φ, μ, η) and (φ, μ, δ). Note that these triangles share the neutral $C = M = Y$ axis with endpoints φ, μ. The goal is to preserve neutrals along the neutral axis (which is a special case of $M + Y = 2C$) axis in the 2-D LUT.

Region 3: Triangles (φ, μ, β) and (φ, μ, γ).

Region 4: Triangles (β, μ, ρ) and (γ, μ, ρ).

Note that the sweeps processed within the four regions described above collectively define the region S_e of control values over which the partial 2-D emulation transform matches the full 4-D emulation transform. To ensure that the points in S_e are representative of values actually sent to emulated device, a suitable amount of K can be added to these sweeps. This can be accomplished by processing the sweeps through a UCR/GCR function appropriate for the emulated device.

C. Performance Analysis

For 8-bit input CMY images, the proposed strategy requires three 2-D LUTs, each LUT is of size $256 \times 511 = 128$ Kbytes. Hence, the overall size is just under 400 Kbytes. The computational processing comprises three additions to compute the s_is and three 2-D table lookups per pixel. This is substantially cheaper than tetrahedral interpolation typically used in 3-D LUTs [14].

As argued in the previous subsection, the traditional tradeoff 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. 8 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 1-D 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. 8) in the Cyan and Magenta 2-D LUTs. With the 2-D approach, this control on device blue does not compromise linearity along the individual separations or gray-balance. With some additional analysis, it may also be possible to define and control the response in other critical regions such as flesh-tones.

The influence of the calibration on points other than those at which the response is specified is illustrated in Fig. 11 for the 2-D Cyan LUT. Three points at which the response is (assumed to be) specified are shown 1) a point $C_0 = M_0 = Y_0$ along the device-gray axis for which a 3-D CIELAB response is specified, 2) a point $C_1 = M_1, Y = 0$ along the device blue axis for which a 2-D CIELAB hue angle and lightness value are specified, and 3) a point $C_0, M = Y = 0$ along the Cyan axis for which a 1-D ΔE from paper response is specified. Note that the 3-D CIELAB response specification for the point $C_0 = M_0 = Y_0$ along the device-gray axis, determines the output value for the Cyan 2-D LUT at the location specified by $C = C_0$, and $M + Y = M_0 + Y_0 = 2M_0 = 2C_0$ (and, also, for the magenta and yellow LUTs at corresponding positions). The same output value is, however, also obtained from the Cyan LUT for all the points along the line formed by the intersection of the planes $M + Y = M_0 + Y_0$ and $C = C_0$. Likewise, the 2-D response specification for the point $C_1 = M_1, Y = 0$ along the device blue axis determines the output value for the Cyan LUT at the location specified by $C = C_1, M + Y = M_1 = C_1$ (and also for the magenta LUT at the corresponding point). The same output value is obtained from the Cyan LUT for all the points along the line formed by the intersection of the planes $M + Y = M_1, C = C_1$. Finally the 1-D response specification for the point $C_0, M = Y = 0$ determines the output value for the Cyan LUT for the point $M + Y = 0, C = C_0$. Note that the same location of the Cyan LUT is not utilized for any other point in input CMY space. For the drawing in Fig. 11, the three points at which the response is specified are related to the point $C_0 = M_0 = Y_0$ on the device gray axis. From the figure, one can see that $C_1 = M_0 + Y_0$. Note, however, that this is not a requirement and was chosen only for convenience in drawing the figure.

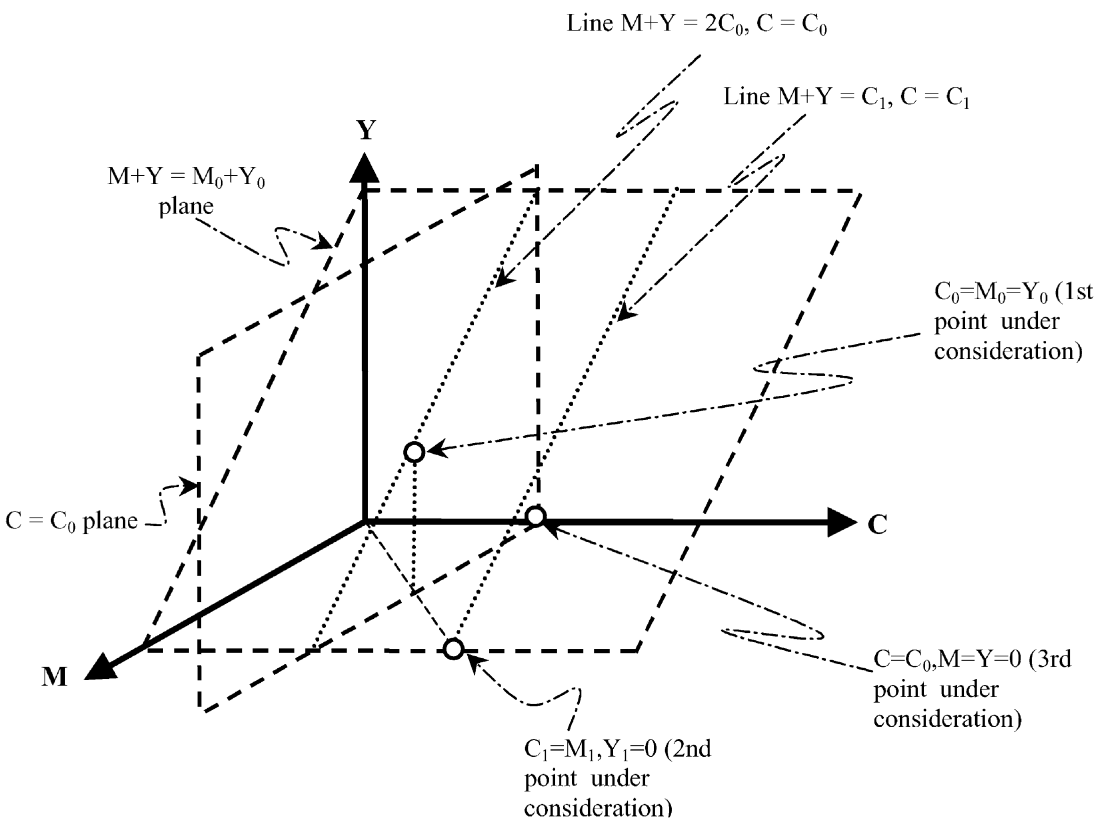


Fig. 11. Illustration of influence of calibration controls outside the emulation set for 2-D calibration as described in Fig. 7.

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 1-D calibration and also explain the greater flexibility offered by the 2-D framework.

VI. Results

A. Device Calibration

An experiment was conducted with a Xerox CMYK laser printer. The 2-D LUTs for C, M, and Y were filled in as described in Section V-B.1. A pure Cyan sweep with 16 levels of $C = 0, 17, 34 \ldots 255, M = 0, Y = 0$ was processed via the two 1-D calibration methods, using 1-D LUTs or TRCs. The first method performed channelwise linearization to ΔE from paper, and the second method used a gray-balance strategy, while linearizing the gray axis to L*. The Cyan sweep was also processed with the 2-D LUTs. Subsequently, the processed sweep was printed and CIELAB values measured. The CIELAB difference from paper white was then computed and plotted (Fig. 12) against increasing value of digital count. A linear response is obtained from the channelwise linearization TRCs, as well as the 2-D correction. In fact, the responses overlap (ignoring measurement noise). This is expected because the pure C, M, and Y axes in the 2-D space spanned by the 2-D LUTs were filled with the respective 1-D channelwise 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.

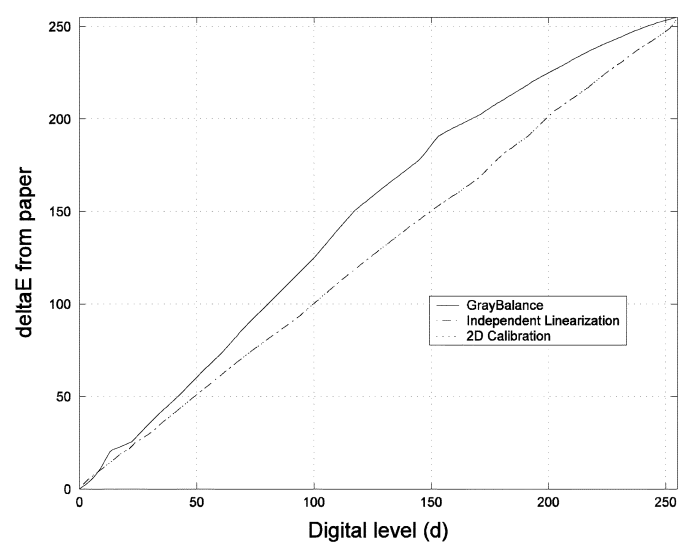


Fig. 12. Plots showing ΔE from paper for the three calibration methods (Cyan channel).

TABLE I
Deviations From True Neutral for the Three Calibration Methods

Calibration Type	Average GB(d)
1-D channel-independent	7.41
1-D gray-balanced	1.17
2-D	1.17

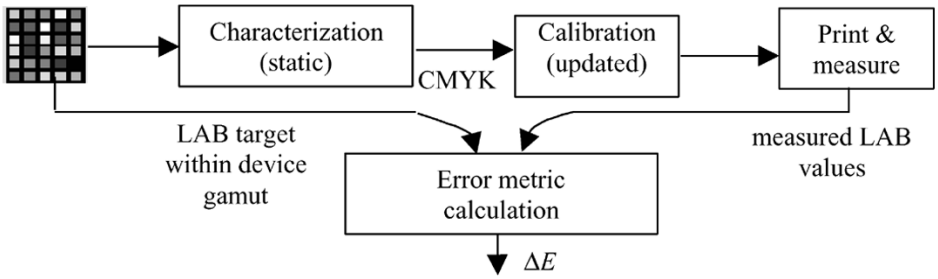


Fig. 13. Stability experiment for evaluating 1-D versus 2-D calibrations.

TABLE II
STABILITY: CIELAB ERRORS MEASURED FOR VARIOUS CALIBRATIONS

Correction	Derived at time:	Measured at time:	Average ΔE^*_{94}	95th %-tile ΔE^*_{94}
1-D gray-balance + characterization	T_0	T_0	2.21	4.08
1-D channel-independent	T_1	T_1	5.78	7.51
1-D gray-balance	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

To evaluate gray reproduction, a $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ sweep was processed via the gray-balance TRC calibration, channelwise calibration, and 2-D calibration, and the resulting outputs printed and measured in CIELAB coordinates. Note that a true neutral sweep would result in $a^* = b^* = 0$ in CIELAB coordinates. Hence, we define a metric similar to the ΔE_{ab} difference as

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

where $a^*(d)$ and $b^*(d)$ are the a^* and b^* components of the CIELAB measurements of a $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ patch with digital value d (as before $d = 0, 17, 34 \ldots 255$). The metric is, hence, defined to quantify the deviations from the true neutral. The average values of $\mathrm{GB}(d)$ for the three calibrations are presented in Table I. The error is significantly larger for the case of channelwise linearization, as compared to gray-balance calibration. This demonstrates that a channelwise linear response does not ensure a gray-balanced response along the $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ axis (we will later address the question as to how stable the gray axis is when a channelwise linearization strategy is used). The errors produced by the gray-balanced TRCs and the 2-D corrections are identical. This is again a natural result of populating the $\mathrm{C} = \mathrm{M} = \mathrm{Y}$ axis in the 2-D C, M, and Y LUTs with the outputs of the 1-D gray-balanced TRCs.

While we have demonstrated the specific case of using ΔE from paper linearization and gray-balance TRCs to fill in the 2-D LUTs, as argued in Section V, it is possible to fill in the other axes (*viz.* white to secondaries, primaries and secondaries to black) differently in order to achieve a desired response along these axes. In general, responses may also be specified along arbitrary curves in the 2-D space of the LUT (i.e., not necessarily straight lines, in Fig. 8) and the rest of the 2-D table can still be easily be filled via some form of 1-D or 2-D interpolation.

B. 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 a Xerox CMYK laser 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 $\mathrm{C'M'Y'K'}$ 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 is repeated for calibrations (1-D ΔE from paper, gray balance and 2-D) derived at time $\mathrm{T}_1 \gg \mathrm{T}_0$, while maintaining the same characterization derived at time T_0. The stability experiment is illustrated in Fig. 13.

Table II 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 II 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 $\mathrm{T}_1 - \mathrm{T}_0$ 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

(a) (b) (c)

Fig. 14. SWOP emulation on Xerox CMYK printer. CMYK image processed via (a) 1-D, (b) 2-D, and (c) 4-D emulation transforms.

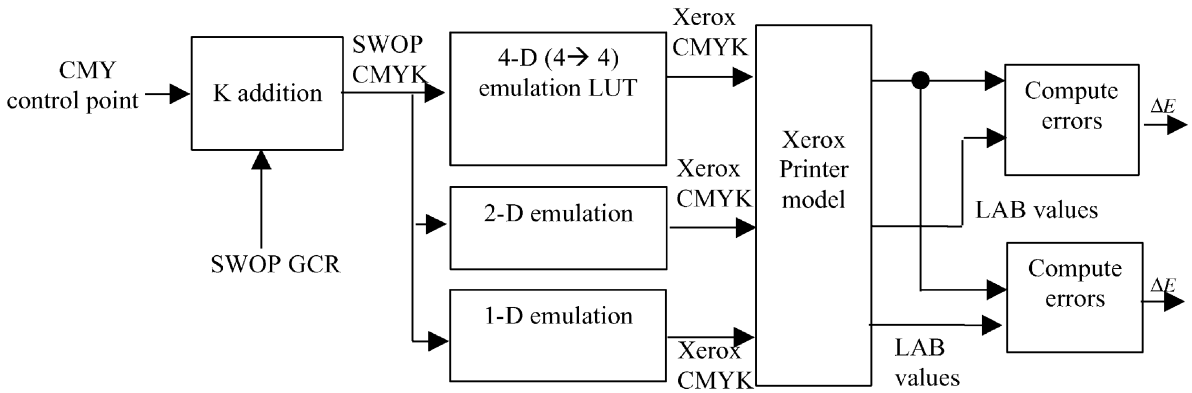


Fig. 15. Schematic diagram of model-based quantitative error evaluation of device emulation.

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, which are, in fact, not much higher than the lower bound. This supports the intuition that 2-D calibration ensures greater stability than 1-D techniques.

C. CMYK *Device Emulation*

We now compare 1-D versus 2-D transforms for device emulation. We consider the emulation of the industry standard SWOP offset CMYK [26] on an experimental Xerox laser CMYK printer. The technique of Section V-B.2 was used to fill the 2-D emulation LUT. A UCR/GCR strategy designed for the SWOP process was used to introduce realistic amounts of K into the emulation process. We designed the 1-D emulation transform so that neutral colors on SWOP would accurately yield neutrals on our experimental Xerox laser printer. In other words, the emulation region $\mathrm{S_e}$ comprises of the set of SWOP CMYK values that map to neutral colors ($a^* = b^* = 0$) in device-independent (CIELAB) coordinates. This approach is, hence, similar to the 1-D gray-balanced calibration described earlier.

1) Qualitative Evaluation: An example of a SWOP (SCID) CMYK image processed via 1-D, 2-D, and 4-D emulation transforms is shown in Fig. 14. It can be visually perceived that 2-D partial emulation offers a closer match to the complete 4-D emulation transform than is afforded by 1-D partial emulation. In other words, the emulated region $\mathrm{S_e}$ is greater for 2-D than for 1-D. This is particularly noticeable in the dark regions in the hair.

2) Quantitative Evaluation: Fig. 15 shows the quantitative evaluation process for comparing 2-D and 1-D emulations against the 4-D emulations. The technique is based on using the 4-D emulation transform [29], [30] as the "ground truth". We generate a set of uniformly distributed CIELAB values well within the SWOP gamut. These values are then mapped through the (inverse) SWOP characterization to generate CMY control values which are subsequently processed through a SWOP GCR strategy to yield SWOP CMYK. The 1-D, 2-D, and 4-D emulation transforms are then used to convert from SWOP CMYK to Xerox CMYK. A Neugebauer printer model for a Xerox CMYK laser printer [20] 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 III 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. We believe this corresponds to significant visual improvement in dark regions in the images.

In addition to model-based evaluation, an empirical test was also conducted by creating color patches with SWOP CMYK values typically exercised in images, and processing them through the 1-D, 2-D, and 4-D emulation transforms. These patches consisted of sweeps from white ($\mathrm{C} = 0, \mathrm{M} = 0, \mathrm{Y} = 0, \mathrm{K} = 0$) to primaries C, M, Y (i.e., $\mathrm{C} = 255, \mathrm{M} = \mathrm{Y} = \mathrm{K} = 0$, etc.) and secondaries R, G and B. Each sweep comprised eight patches equally spaced between 0

TABLE III
QUANTITATIVE EVALUATION OF 1-D VERSUS 2-D EMULATION USING PRINTER MODEL

Emulation Type	Average ΔE_{94} error	95 % ΔE_{94} error	Max. ΔE_{94} error
1-D emulation	5.8714	17.2145	22.8000
2-D emulation	1.9922	11.1413	12.2488

TABLE IV
NEAR-NEUTRAL SWOP CMYK VALUES USED IN THE MEASUREMENT-BASED QUANTITATIVE EVALUATION OF 2-D EMULATION, DESCRIBED IN SECTION VI-C.2

C	M	Y	K
23	15	1	0
57	42	25	0
115	86	58	0
170	134	118	27
222	194	174	102
255	237	218	150
255	255	255	237
1	1	1	0
25	25	25	0
58	58	58	0
118	118	118	27
174	174	174	102
218	218	218	150

TABLE V
QUANTITATIVE EVALUATION OF 2-D VERSUS 1-D EMULATION USING MEASURED DATA

Emulation Type	Average ΔE_{94} error	95 % ΔE_{94} error	Max. ΔE_{94} error
1-D emulation	4.7825	14.5586	16.5688
2-D emulation	2.2873	9.83	12.374

and 255. In addition, a set of near-neutral patches were included, whose values are tabulated in Table IV. The processed patches were printed and measured to obtain three sets of CIELAB values corresponding to the 1-D, 2-D, and 4-D emulation transforms. As before, the CIE94 errors were computed between the 1-D (or 2-D) and 4-D CIELAB values. Table V shows the CIE94 errors computed for the empirical test. Once again, we note the superior performance of the 2-D technique.

It is to be noted that the approach of filling in the 2-D emulation LUTs by mimicking the 4-D response along certain projections is significantly different from the technique used in the calibration application, where certain 1-D loci are specified and the rest of the 2-D space is populated by linear interpolation. Our experiments indicated that simple linear interpolation between 1-D loci was not satisfactorily accurate for the emulation application. This is because, unlike the calibration case, the function being approximated is highly nonlinear, resulting in much greater tensions among the responses along the various 1-D loci. To illustrate this, the K channel for the Xerox laser printer is plotted in Fig. 16(a) as a function of SWOP K and min(C, M, Y) (i.e., the domain of the 2-D LUT for the K channel). The graph was obtained by processing various combinations of SWOP $C = M = Y$ and SWOP K through the 4-D emulation transform, and plotting the resulting Xerox printer K value. Note the highly nonlinear nature of the 4-D emulation function. Fig. 16(b) and (c) show the 1-D and 2-D approximations to the emulation transform. The surface plots in Fig. 16(a) and (c) are, in fact, identical. This is to be expected, since the 4-D and 2-D responses were designed to agree within

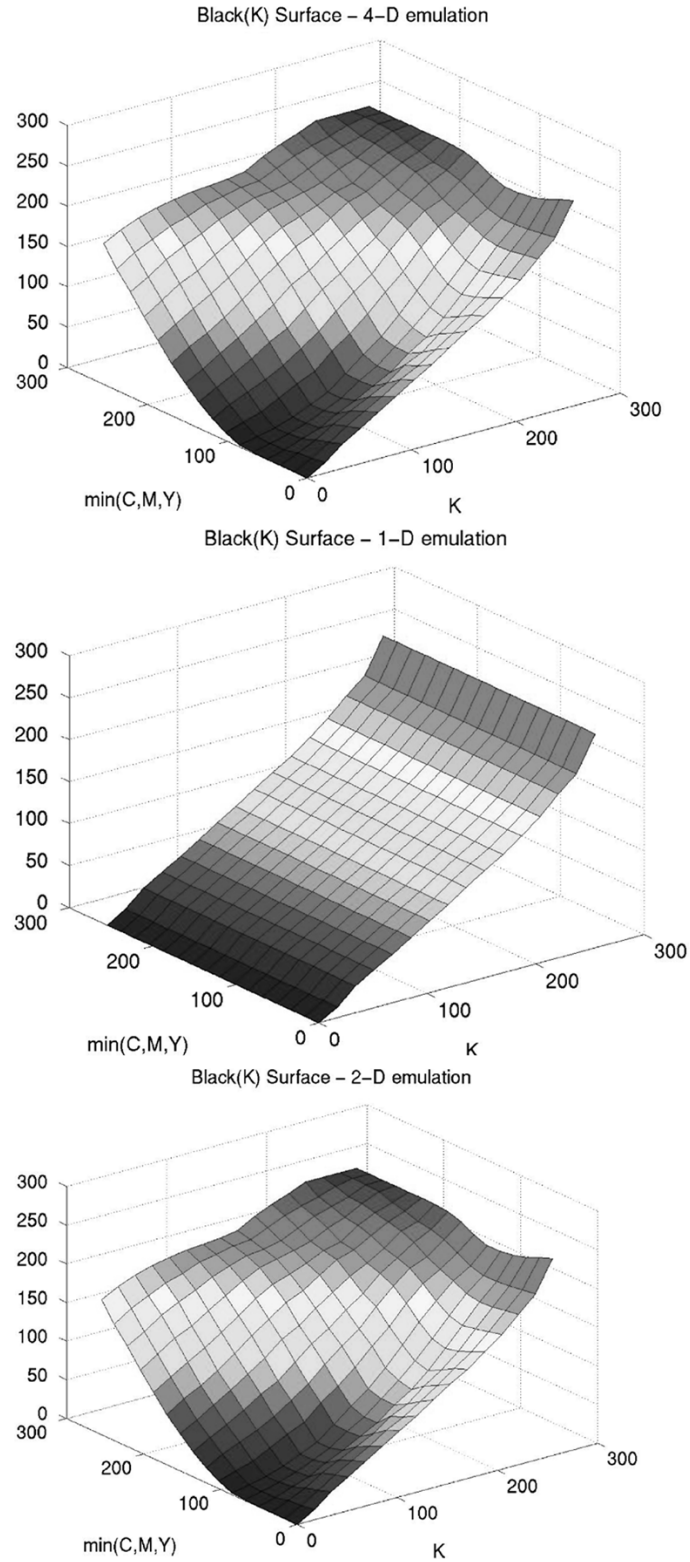


Fig. 16. Output K as a function of SWOP K versus min(C, M, Y) (a) 4-D, (b) 1-D, and (c) 2-D emulation.

the chosen domain of the 2-D LUT for K. In Fig. 16(a), a high value of Xerox laser K is seen for high SWOP CMY (and SWOP $K = 0$). In Fig. 16(b), however, input $K = 0$ always maps to output $K = 0$. The 1-D K TRC is a monotonic function of K alone, and does not account for interactions with C, M, Y. A similar behavior may be observed by comparing the cyan surfaces for 1-D, 2-D and 4-D emulation. For dark colors (i.e., high-input SWOP CMY), the 4-D and 2-D emulations would yield low CMY amounts and large K as is necessary to reproduce these colors on the particular experimental Xerox CMYK laser printer. However, the monotonic 1-D emulation transform cannot exhibit this characteristic and, hence, introduces significant errors in the dark regions.

VII. Conclusion

We present a general framework for the color correction of output devices that unifies device calibration and device emulation. Emulation is the general problem of mimicking the response of one device on another. Calibration is a special case of emulation, wherein the device being emulated obeys mathematically defined ideal characteristics. Within this framework, a novel 2-D 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 a real device with another. Through experimental results we demonstrate that, for both calibration and emulation, the 2-D scheme offers far superior quality compared to the 1-D equivalent, while incurring a relatively modest computational overhead that is significantly lower than that required by 3-D/4-D methods, and that is likely to be acceptable for real-time processing of color images. Finally, though our results are primarily obtained for printing devices, the concepts can be readily applied to additive color devices, in particular for LCD displays where the conventional 1-D calibration imposes similar undesirable constraints.

Acknowledgment

The authors would like to thank Dr. V. Klassen for his valuable comments and suggestions on this work. They would also like to thank the reviewers for their assistance in improving the quality of this paper.

References

[1] R. Bala, *Device Characterization, Chapter 5 in Digital Color Imaging Handbook*, G. Sharma, Ed. Boca Raton, FL: CRC, 2003.
[2] E. J. Giorgianni, T. E. Madden, and K. E. Spaulding, *Color Management for Digital Imaging Systems, Chapter 4 in Digital Color Imaging Handbook*, G. Sharma, Ed. Boca Raton, FL: CRC, 2003.
[3] E. J. Giorgianni and T. E. Madden, *Digital Color Management: Encoding Solutions*. Reading, MA: Addison Wesley, 1998.
[4] CIE, Colorimetry, CIE Pub. no. 15.2, Centr. Bureau CIE, Vienna, Austria, 1986.
[5] G. Sharma, Ed., *Color Fundamentals for Digital Imaging, Chapter 1 in Digital Color Imaging Handbook*. Boca Raton, FL: CRC, 2003.
[6] R. W. G. Hunt, *Measuring Color*. Upper Saddle River, NJ: Prentice-Hall, 1991.
[7] International Color Consortium [Online]. Available: www.color.org
[8] G. Sharma and H. J. Trussell, "Digital color imaging," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 901–932, Jul. 1997.
[9] G. Sharma, M. J. Vrhel, and H. J. Trussell, "Color imaging for multimedia," *Proc. IEEE*, vol. 86, no. 6, pp. 1088–1108, Jun. 1998.
[10] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed. New York: Wiley, 1982.
[11] M. J. Vrhel and H. J. Trussell, "Color device calibration: A mathematical formulation," *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1796–1806, Dec. 1999.
[12] V. Ostromoukhov, R. D. Hersch, C. Perarie, P. Emmel, and I. Amidror, "Two approaches in scanner-printer calibration: Colorimetric space based vs. closed-loop," in *Proc. SPIE/IS&T Int. Symp. Electronic Imaging*, vol. 2170, San Jose, CA, 1994, pp. 133–142.
[13] Y. Wu, "Combine 1-D and 3-D color calibration methods for ensuring consistent color reproduction," in *Proc. SPIE: Color Imaging IX: Processing Hardcopy, and Applications*, vol. 5293, San Jose, CA, 2004, pp. 242–259.
[14] R. Bala and R. V. Klassen, *Efficient Color Transformation Implementation, Chapter 11 in Digital Color Imaging Handbook*, G. Sharma, Ed. Boca Raton, FL: CRC, 2003.
[15] J. Z. Chang, J. P. Allebach, and C. A. Bouman, "Sequential linear interpolation of multidimensional functions," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1231–1245, Sep. 1997.
[16] H. R. Kang, *Color Technology for Electronic Imaging Devices*. Bellingham, WA: SPIE, 1997.
[17] P. C. Hung, "Colorimetric calibration in electronic imaging devices using a look-up table model and interpolations," *J. Electron. Imag.*, vol. 2, no. 1, pp. 53–61, 1993.
[18] J. M. Kassen, S. I. Nin, W. Plouffe, and J. L. Hafner, "Performing color space conversions with three-dimensional linear interpolation," *J. Electron. Imag.*, vol. 4, no. 3, pp. 226–250, Jul. 1995.
[19] R. Bala, V. Monga, G. Sharma, and J. P. V. Capelle, "Two-dimensional transforms for device color calibration," in *Proc. SPIE: Color Imaging IX: Processing Hardcopy, and Applications*, vol. 5293, San Jose, CA, 2004, pp. 250–261.
[20] R. Bala, "Optimization of the spectral Neugebauer model for printer characterization," *J. Electron. Imag.*, vol. 8, no. 2, pp. 156–166, 1999.
[21] R. Balasubramanian, "Colorimetric models for binary printers," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 1995, pp. II-327–II-330.
[22] M. Xia, E. Saber, G. Sharma, and A. M. Tekalp, "End-to-end color printer calibration by total least squares regression," *IEEE Trans. Image Process.*, vol. 8, no. 5, pp. 700–716, May 1999.
[23] A. U. Agar and J. P. Allebach, "A minimax method for function interpolation using an SLI structure," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 1997, pp. 671–674.
[24] R. Balasubramanian, "Refinement of printer transformations using weighted regression," *Proc. SPIE*, vol. 2658, pp. 334–340, 1996.
[25] Y. X. Noyes, J. Y. Hardeberg, and A. Moskalev, "Linearization curve generation for CcMmYK printing," in *Proc. IS&T and SIDs 8th Color Imaging Conf.*, Nov. 2000, pp. 247–251.
[26] Specifications web offset publications [Online]. Available: www.swop.org
[27] What is sRGB—Introduction to the standard default RGB color space developed by Hewlett-Packard and Microsoft [Online]. Available: http://www.srgb.com/aboutsrgb.html
[28] R. Hunt, "Why is black-and-white so important in color?," in *Proc. IS&T and SIDs 4th Color Imaging Conf.*, Scottsdale, AZ, Nov. 1996, pp. 54–57.
[29] T. J. Cholewo, "Conversion between CMYK spaces preserving black separation," in *Proc. IS&T/SIDs8th Color Imaging Conf.*, Nov. 2000, pp. 257–261.
[30] H. Zeng, "CMYK transformation with black preservation in color management system," in *Proc. SPIE Annu. Symp. Electronic Imaging*, vol. 4663, 2002, pp. 143–149.

Raja Bala received the B.S. degree from the University of Texas, Arlington, in 1987, and the M.S. and Ph.D. degrees from Purdue University, West Lafayette, IN, in 1988 and 1992, respectively, all in electrical engineering. His Ph.D dissertation was on efficient algorithms for color quantization of images.

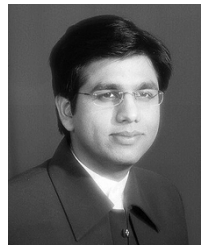
He is currently with the Xerox Innovation Group, Webster, NY, where he is a Principal Scientist involved in a project on color science and color management. He is also an Adjunct Professor with the School of Electrical Engineering, Rochester Institute of Technology, Rochester, NY. He holds over 30 patents and is the author of over 40 publications in the field of color imaging.

Dr. Bala is a member of IS&T

Gaurav Sharma (SM'00) received the B.E. degree in electronics and communication engineering from the Indian Institute of Technology (formerly University of Roorkee), Roorkee, in 1990, the M.E. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1992, and the M.S. degree in applied mathematics and the Ph.D. degree in electrical and computer engineering from North Carolina State University (NCSU), Raleigh, in 1995 and 1996, respectively.

From August 1992 to August 1996, he was a Research Assistant at the Center for Advanced Computing and Communications, Electrical and Computer Engineering Department, NCSU. From August 1996 to August 2003, he was with Xerox Research and Technology, Webster, NY, initially as a member of research staff and subsequently at the position of Principal Scientist. Since Fall 2003, he has been an Associate Professor at the University of Rochester, Rochester, NY. His research interests include multimedia security and watermarking, color science and imaging, signal restoration, and halftoning.

Dr. Sharma is a member of Sigma Xi, Phi Kappa Phi, Pi Mu Epsilon, and IS&T. He was the 2003 Chair for the Rochester chapter of the IEEE Signal Processing Society and currently serves as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and the SPIE/IS&T *Journal of Electronic Imaging*.



Vishal Monga (S'00–M'01) received the B.Tech degree in electrical engineering from the Indian Institute of Technology (IIT), Guwahati, in May 2001, and the M.S.E.E. and Ph.D. degrees from The University of Texas, Austin, in May 2003 and August 2005, respectively.

His research interests span signal and image processing, applied linear algebra, and information theory. He has researched problems in color imaging, particularly halftoning and device color correction, information embedding in multimedia, and perceptual media hashing.

Dr. Monga is a member of SPIE and IS&T. He received the IS&T Raymond Davis scholarship in 2004, a Texas Telecommunications Consortium (TxTec) Graduate Fellowship from The University of Texas for the year 2002 to 2003, and the President's Silver Medal in 2001 from IIT, Guwahati.

Jean-Pierre Van de Capelle, photograph and biography not available at the time of publication.