chapter five

Device characterization

Raja Balasubramanian
Xerox Solutions & Services Technology Center

Contents

5.1 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 understanding is achieved through a process of device characterization. One approach for doing this is known as closed-loop characterization, where a specific input device is optimized for rendering images to a specific output device. A common example of closed-loop systems is found in offset press printing, where a drum scanner is often tuned to output CMYK signals for optimum reproduction on a particular offset press. The tuning is often carried out manually by skilled press operators. Another example of a closed-loop system is traditional photography, where the characteristics of the photographic dyes, film, development, and printing processes are co-optimized (again, often manually) for proper reproduction. While the closed-loop paradigm works well in the aforementioned examples, it is not an efficient means of managing color in open digital color imaging systems where color can be exchanged among a large and variable number of color devices. For example, a system comprising three scanners and four printers would require $3 \times 4 = 12$ closed-loop transformations. Clearly, as more devices are added to the system, it becomes difficult to derive and maintain characterizations for all the various combinations of devices.

An alternative approach that is increasingly embraced by the digital color imaging community is the device-independent paradigm, where translations among different device color representations are accomplished via an intermediary device-independent color representation. This approach is more efficient and easily managed than the closed-loop model. Taking the same example of three scanners and four printers now requires only $3 + 4 = 7$ transformations. The device-independent color space is usually based on a colorimetric standard such as CIE XYZ or CIELAB. Hence, the visual system is explicitly introduced into the color imaging path. The closed-loop and device-independent approaches are compared in Figure 5.1.

The characterization techniques discussed in this chapter subscribe to the device-independent paradigm and, as such, involve deriving transformations between device-dependent and colorimetric representations. Indeed, a plethora of device characterization techniques have been reported in the literature. The optimal approach depends on several factors, including the physical color characteristics of the device, the desired quality of the characterization, and the cost and effort that one is willing to bear to perform the characterization. There are, however, some fundamental concepts that are common to all these approaches. We begin this chapter with a description of these concepts and then provide a more detailed exposition of characterization techniques for commonly encountered input and output devices. To keep the chapter to a manageable size, an exhaustive treatment is given to only a few topics. The chapter is complemented by an extensive set of references for a more in-depth study of the remaining topics.

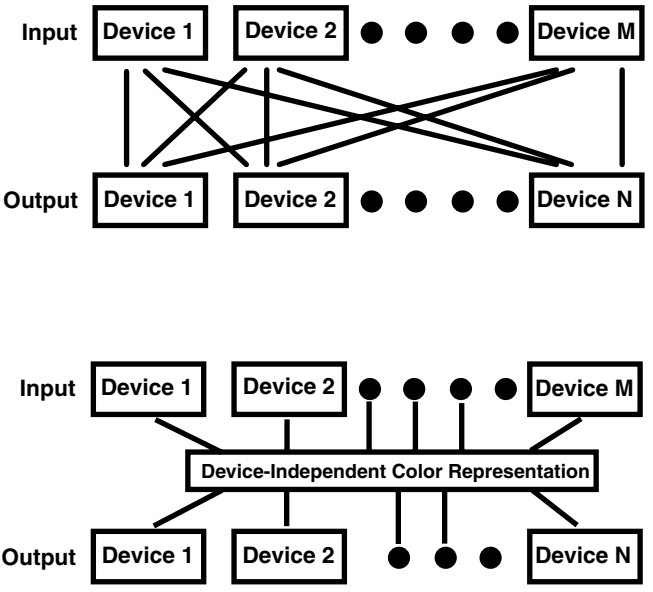


Figure 5.1 Closed-loop vs. device-independent color management.

5.2 Basic concepts

It is useful to partition the transformation between device-dependent and device-independent space into a calibration and a characterization function, as shown in Figure 5.2.

5.2.1 Device calibration

Device calibration is the process of maintaining the device with a fixed known characteristic color response and is a precursor to characterization. Calibration can involve simply ensuring that the controls internal to the device are kept at fixed nominal settings (as is often the case with scanners and digital cameras). Often, if a specific color characteristic is desired, this typically requires making color measurements and deriving correction functions to ensure that the device maintains that desired characteristic. Sometimes the desired characteristic is defined individually for each of the device signals; e.g., for a CRT display, each of the R, G, B channels is often linearized with respect to luminance. This linearization can be implemented with a set of one-dimensional tone reproduction curves (TRCs) for each of the R, G, B signals. Sometimes, the desired characteristic is defined in terms of mixtures of device signals. The most common form of this is gray-balanced calibration, whereby equal amounts of device color signals (e.g., R = G = B or C = M = Y) correspond to device-independent measurements that are neutral or gray

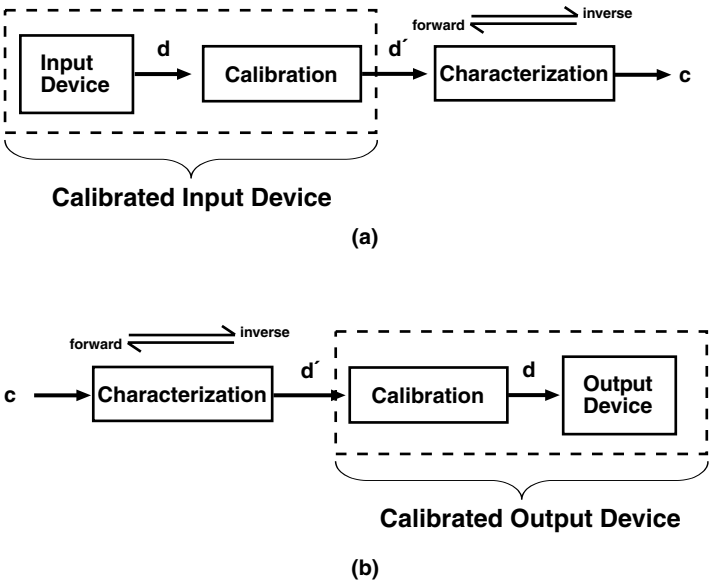


Figure 5.2 Calibration and characterization for input and output devices.

(e.g., a*= b*= 0 in CIELAB coordinates). Gray-balancing of a device can also be accomplished with a set of TRCs.

It is important to bear mind that calibration with one-dimensional TRCs can control the characteristic response of the device only in a limited region of color space. For example, TRCs that ensure a certain tone response along each of the R, G, B axes do not necessarily ensure control of the gray axis, and vice versa. However, it is hoped that this limited control is sufficient to maintain, within a reasonable tolerance, a characteristic response within the entire color gamut; indeed, this is true in many cases.

5.2.2 Device characterization

The characterization process derives the relationship between device-dependent and device-independent color representations for a calibrated device. For input devices, the captured device signal is first processed through a calibration function (see Figure 5.2) while output devices are addressed through a final calibration function. In typical color management workflows, device characterization is a painstaking process that is done infrequently, while the simpler calibration process is carried out relatively frequently to compensate for temporal changes in the device's response and maintain it in a fixed known state. It is thus assumed that a calibrated device maintains the validity of the characterization function at all times. Note that calibration and characterization form a pair, so that if a new calibration alters the characteristic color response of the device, the characterization must also be re-derived.

The characterization function can be defined in two directions. The forward characterization transform defines the response of the device to a known input, thus describing the color characteristics of the device. The inverse characterization transform compensates for these characteristics and determines the input to the device that is required to obtain a desired response. The inverse function is used in the final imaging path to perform color correction to images.

The sense of the forward function is different for input and output devices. For input devices, the forward function is a mapping from a device-independent color stimulus to the resulting device signals recorded when the device is exposed to that stimulus. For output devices, this is a mapping from device-dependent colors driving the device to the resulting rendered color, in device-independent coordinates. In either case, the sense of the inverse function is the opposite to that of the forward function.

There are two approaches to deriving the forward characterization function. One approach uses a model that describes the physical process by which the device captures or renders color. The parameters of the model are usually derived with a relatively small number of color samples. The second approach is empirical, using a relatively large set of color samples in conjunction with some type of mathematical fitting or interpolation technique to derive the characterization function. Derivation of the inverse function calls for an empirical or mathematical technique for inverting the forward function. (Note that the inversion does not require additional color samples; it is purely a computational step.)

A primary advantage to model-based approaches is that they require fewer measurements and are thus less laborious and time consuming than empirical methods. To some extent, a physical model can be generalized for different image capture or rendering conditions, whereas an empirical technique is typically optimized for a restrictive set of conditions and must be rederived as the conditions change. Model-based approaches generate relatively smooth characterization functions, whereas empirical techniques are subject to additional noise from measurements and often require additional smoothing on the data. However, the quality of a model-based characterization is determined by the extent to which the model reflects the real behavior of the device. Certain types of devices are not readily amenable to tractable physical models; thus, one must resort to empirical approaches in these cases. Also, most model-based approaches require access to the raw device, while empirical techniques can often be applied in addition to simple calibration and characterization functions already built into the device. Finally, hybrid techniques can be employed that borrow strengths from both model-based and empirical approaches. Examples of these will be presented later in the chapter.

The output of the calibration and characterization process is a set of mappings between device-independent and -dependent color descriptions; these are usually implemented as some combination of power-law mapping, 3×3 matrix conversion, white-point normalization, and one-dimensional and multidimensional lookup tables. This information can be stored in a

variety of formats, of which the most widely adopted industry standard is the International Color Consortium (ICC) profile (www.color.org). For printers, the Adobe Postscript language (Level 2 and higher) also contains operators for storing characterization information.[1]

It is important to bear in mind that device calibration and characterization, as described in this chapter, are functions that depend on color signals alone and are not functions of time or the spatial location of the captured or rendered image. The overall accuracy of a characterization is thus limited by the ability of the device to exhibit spatial uniformity and temporal stability. Indeed, in reality, the color characteristics of any device will vary to some degree over its spatial footprint and over time. It is generally good practice to gather an understanding of these variances prior to or during the characterization process. This may be accomplished by exercising the device response with multiple sets of stimuli in different spatial orientations and over a period of time. The variation in the device's response to the same stimulus across time and space is then observed. A simple way to reduce the effects of nonuniformity and instability during the characterization process is to average the data at different points in space and time that correspond to the same input stimulus.

Another caution to keep in mind is that many devices have color-correction algorithms already built into them. This is particularly true of low-cost devices targeted for consumers. These algorithms are based in part on calibration and characterization done by the device manufacturer. In some devices, particularly digital cameras, the algorithms use spatial context and image-dependent information to perform the correction. As indicated in the preceding paragraph, calibration or characterization by the user is best performed if these built-in algorithms can be deactivated or are known to the extent that they can be inverted. (This is especially true of the model-based approaches.) Reverse engineering of built-in correction functions is not always an easy task. One can also argue that, in many instances, the built-in algorithms provide satisfactory quality for the intended market, hence not requiring additional correction. Device calibration and characterization is therefore recommended only when it is necessary and possible to fully control the color characteristics of the device.

5.2.3 *Input device calibration and characterization*

There are two main types of digital color input devices: scanners, which capture light reflected from or transmitted through a medium, and digital cameras, which directly capture light from a scene. The captured light passes through a set of color filters (most commonly, red, green, blue) and is then sensed by an array of charge-coupled devices (CCDs). The basic model that describes the response of an image capture device with M filters is given by

$$D_i = \int_{\lambda \varepsilon V} S(\lambda) q_i(\lambda) u(\lambda) \partial\lambda + n_i, i = 1, \ldots, M \tag{5.1}$$

where D_i = sensor response
$S(\lambda)$ = input spectral radiance
$q_i(\lambda)$ = spectral sensitivity of the ith sensor
$u(\lambda)$ = detector sensitivity
n_i = measurement noise in the ith channel
V = spectral regime outside which the device sensitivity is negligible

Digital still cameras often include an infrared (IR) filter; this would be incorporated into the $u(\lambda)$ term. Invariably, $M = 3$ sensors are employed with filters sensitive to the red, green, and blue portions of the spectrum. The spectral sensitivities of a typical set of scanner filters are shown in Figure 5.3. Scanners also contain an internal light source that illuminates the reflective or transmissive material being scanned. Figure 5.4 shows the spectral radiance of a fluorescent scanner illuminant. Note the sharp spikes that typify fluorescent sources. The light incident upon the detector is given by

$$S(\lambda) = I_s(\lambda)R(\lambda) \tag{5.2}$$

where $R(\lambda)$ = spectral reflectance (or transmittance) function of the input stimulus
$I_s(\lambda)$ = scanner illuminant

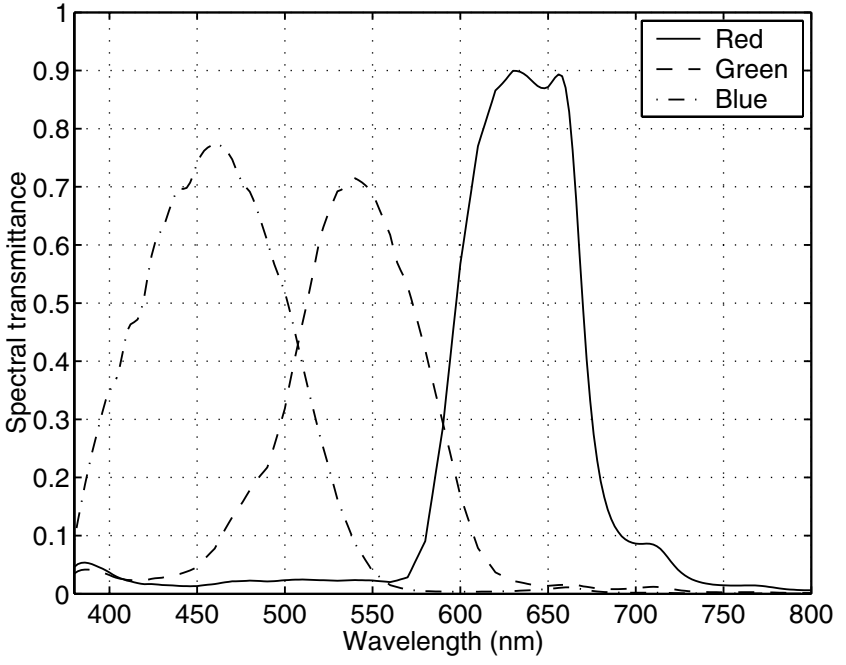


Figure 5.3 Typical scanner filter sensitivities.

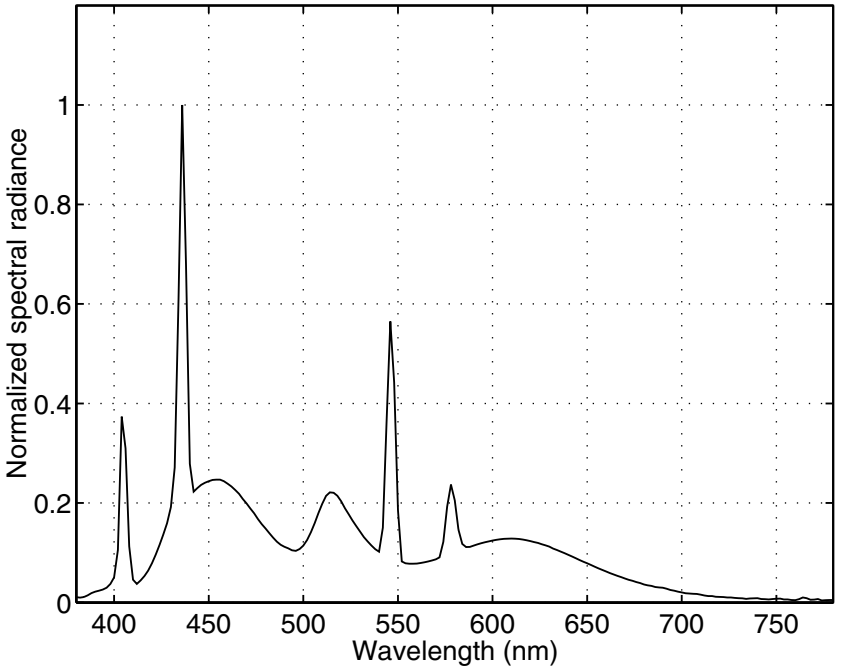


Figure 5.4 Spectral radiance of typical scanner illuminant.

From the introductory chapter on colorimetry, we know that spectral radiance is related to colorimetric signals by

$$C_i = K_i \int_{\lambda \varepsilon V} S(\lambda) c_i(\lambda) \partial\lambda, i = 1,2,3 \tag{5.3}$$

where C_i = colorimetric signals
$c_i(\lambda)$ = corresponding color matching functions
K_i = normalizing constants

Again, if a reflective sample is viewed under an illuminant $I_v(\lambda)$, the input spectral radiance is given by

$$S(\lambda) = I_v(\lambda) R(\lambda) \tag{5.4}$$

Equations 5.1 through 5.4 together establish a relationship between device-dependent and device-independent signals for an input device. To further explore this relationship, let us represent a spectral signal by a discrete L-vector comprising samples at wavelengths $\lambda_1, \ldots, \lambda_L$. Equation 5.1 can be rewritten as

$$\mathbf{d} = A_d^t \mathbf{s} + \varepsilon \tag{5.5}$$

where $\mathbf{d}$ = M-vector of device signals
$\mathbf{s}$ = L-vector describing the input spectral signal
$\mathbf{A}_d$ = $L \times M$ matrix whose columns are the input device sensor responses
ε = noise term

If the input stimulus is reflective or transmissive, then the illuminant term $I_s(\lambda)$ can be combined with either the input signal vector $\mathbf{s}$ or the sensitivity matrix $\mathbf{A}_d$. In a similar fashion, Equation 5.3 can be rewritten as

$$\mathbf{c} = \mathbf{A}_c^t \mathbf{s} \tag{5.6}$$

where $\mathbf{c}$ = colorimetric three-vector
$\mathbf{A}_c$ = $L \times 3$ matrix whose columns contain the color-matching functions $c_i(\lambda)$

If the stimulus being viewed is a reflection print, then the viewing illuminant $I_v(\lambda)$ can be incorporated into either $\mathbf{s}$ or $\mathbf{A}_c$.

It is easily seen from Equations 5.5 and 5.6 that, in the absence of noise, a unique mapping exists between device-dependent signals $\mathbf{d}$ and device-independent signals $\mathbf{c}$ if there exists a transformation from the device sensor response matrix $\mathbf{A}_d$ to the matrix of color matching functions $\mathbf{A}_c$.[2] In the case of three device channels, this translates to the condition that $\mathbf{A}_d$ must be a linear nonsingular transformation of $\mathbf{A}_c$.[3,4] Devices that fulfill this so-called *Luther–Ives* condition are referred to as *colorimetric* devices.

Unfortunately, practical considerations make it difficult to design sensors that meet this condition. For one thing, the assumption of a noise-free system is unrealistic. It has been shown that, in the presence of noise, the Luther–Ives condition is not optimal in general, and it guarantees colorimetric capture only under a single viewing illuminant I_v.[5] Furthermore, to maximize the efficiency, or signal-to-noise ratio (SNR), most filter sets are designed to have narrowband characteristics, as opposed to the relatively broadband color matching functions. For scanners, the peaks of the R, G, B filter responses are usually designed to coincide with the peaks of the spectral absorption functions of the C, M, Y colorants that constitute the stimuli being scanned. Such scanners are sometimes referred to as *densitometric* scanners. Because photography is probably the most common source for scanned material, scanner manufacturers often design their filters to suit the spectral characteristics of photographic dyes. Similar observations hold for digital still cameras, where filters are designed to be narrowband, equally spaced, and independent so as to maximize efficiency and enable acceptable shutter speeds. A potential outcome of this is scanner metamerism, where two

stimuli that appear identical to the visual system may result in distinct scanner responses, and vice versa.

The spectral characteristics of the sensors have profound implications on input device characterization. The narrowband sensor characteristics result in a relationship between XYZ and device RGB that is typically more complex than a 3×3 matrix, and furthermore changes as a function of properties of the input stimulus (i.e., medium, colorants, illuminant). A colorimetric filter set, on the other hand, results in a simple linear characterization function that is media independent and that does not suffer from metamerism. For these reasons, there has been considerable interest in designing filters that approach colorimetric characteristics, subject to practical constraints that motivate the densitometric characteristics.[6] An alternative approach is to employ more than three filters to better approximate the spectral content of the input stimulus.[7] These efforts are largely in the research phase; most input devices in the market today still employ three narrowband filters. Hence, the most accurate characterization is a nonlinear function that varies with the input medium.

Model-based characterization techniques use the basic form of Equation 5.1 to predict device signals D_i given the radiance $S(\lambda)$ of an arbitrary input medium and illuminant, and the device spectral sensitivities. The latter can sometimes be directly acquired from the manufacturer. However, due to temporal changes in device characteristics and variations from device to device, a more reliable method is to estimate the sensitivities from measurements of suitable targets. Model-based approaches may be used in situations where there is no way of determining *a priori* the characteristics of the specific stimulus being scanned. However, the accuracy of the characterization is directly related to the accuracy of the model and its estimated parameters. The result is usually an $M \times 3$ matrix that maps M (typically three) device signals to three colorimetric signals such as XYZ.

Empirical techniques, on the other hand, directly correlate colorimetric measurements of a color target with corresponding device values that result when the device is exposed to the target. Empirical techniques are suitable when the physical nature of the input stimulus is known beforehand, and a color target with the same physical traits is available for characterizing the input device. An example is the use of a photographic target to characterize a scanner that is expected to scan photographic prints. The characterization can be a complex nonlinear function chosen to achieve the desired level of accuracy, and it is obtained through an empirical data-fitting or interpolation procedure.

Modeling techniques are often used by researchers and device manufacturers to better understand and optimize device characteristics. In end user applications, empirical approaches are often adopted, as these provide a more accurate characterization than model-based approaches for a specific set of image capture conditions. This is particularly true for the case of scanners, where it is possible to classify *a priori* a few commonly encountered media (e.g., photography, lithography, xerography, inkjet) and generate

empirical characterizations for each class. In the case of digital cameras, it is not always easy to define or classify the type of stimuli to be encountered in a real scene. In this case, it may be necessary to revert to model-based approaches that assume generic scene characteristics. More details will be presented in following sections.

A generic workflow for input device characterization is shown in Figure 5.5. First, the device is calibrated, usually by ensuring that various internal settings are in a fixed nominal state. For scanners, calibration minimally involves normalizing the RGB responses to the measurement of a built-in white tile, a process that is usually transparent to the user. In addition, it may be desirable to linearize and gray-balance the device response by scanning a suitable premeasured target. Next, the characterization is performed using a target comprising a set of color patches that spans the gamut of the input medium. Often, the same target is used for both linearization and characterization. Industry standard targets designed for scanners are the Q60 and IT8. Device-independent color measurements are made of each patch in the target using a spectroradiometer, spectrophotometer, or colorimeter. Additional data processing may be necessary to extract raw colorimetric data from the measurements generated by the instrument. Next, the input device records an image of the target. If characterization is being performed as a separate step after calibration, then the captured image must be processed through the calibration functions derived in a previous step. The device-dependent (typically RGB) coordinates for each patch on the target must then be extracted from the image. This involves correctly identifying the spatial extent of each patch within the scanned image. To facilitate this, it is desirable to include reference fiducial marks at each corner of the target and supply target layout information (e.g., number of rows, columns) to the image-processing software. Also, it is recommended that a subset of pixels near the center of each patch is averaged, so as to reduce the effect of spatial noise in the device response. Once extracted, the device-dependent values are correlated with the corresponding device-independent values to obtain the characterization for the device.

The forward characterization is a model of how the device responds to a known device-independent input; i.e., it is a function that maps device-

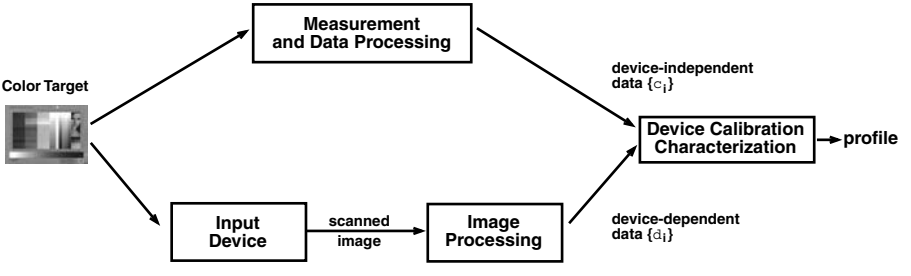


Figure 5.5 Input device characterization workflow.

independent measurements to the resulting device signals. The inverse function compensates for the device characteristics and maps device signals to corresponding device-independent values. Model-based techniques estimate the forward function, which is then inverted using analytic or numerical approaches. Empirical techniques derive both the forward and inverse functions.

Figure 5.6 describes how the accuracy of the resulting characterization can be evaluated. A test target containing colors that are preferably different from those in the initial characterization target is presented to the image-capture device. The target should be made with the same colorants and media as used for the characterization target. The resulting captured electronic image is mapped through the same image-processing functions performed when the characterization was derived (see Figure 5.5). It is then converted to a device-independent color space using the inverse characterization function. The device-independent color values of the patches are then extracted and compared with measurements of these patches using an appropriate color difference formula such as ΔE^*_{ab} or ΔE^*_{94} (described in more detail in Section 5.5). To avoid redundant processing, the same target can be used for both deriving and testing the characterization, with different portions of the target being used for the two purposes.

5.2.4 Output device calibration and characterization

Output color devices can be broadly categorized into emissive display devices and devices that produce reflective prints or transparencies. Emissive devices produce colors via additive mixing of red, green, and blue (RGB) lights. Examples are cathode ray tube (CRT) displays, liquid crystal displays (LCDs), organic light emitting diodes (OLEDs), plasma displays, projection displays, etc. The spectral radiance emitted by a display device is a function of the input digital RGB values and is denoted $S_{RGB}(\lambda)$. Two important assumptions are usually made that greatly simplify display characterization.

- *Channel independence.* Each of the R, G, B channels to the display operates independently of the others. This assumption allows us to separate the contribution of spectral radiance from the three channels.

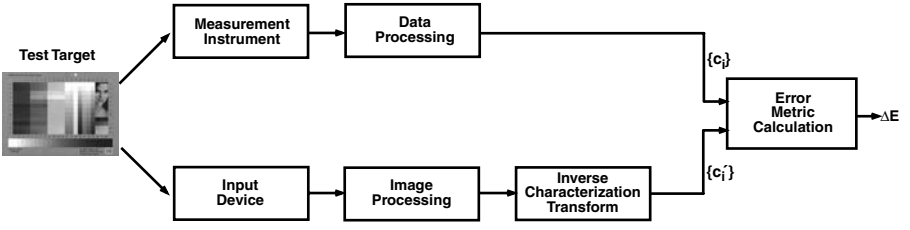


Figure 5.6 Testing of input device characterization.

$$S_{RGB}(\lambda) = S_R(\lambda) + S_G(\lambda) + S_B(\lambda) \quad (5.7)$$

- *Chromaticity constancy.* The spectral radiance due to a given channel has the same basic shape and is only scaled as a function of the device signal driving the display. This assumption further simplifies Equation 5.7 to

$$S_{RGB}(\lambda) = f_R(D_R)\, S_{Rmax}(\lambda) + f_G(D_G)\, S_{Gmax}(\lambda) + f_B(D_B)\, S_{Bmax}(\lambda) \quad (5.8)$$

where $S_{Rmax}(\lambda)$ = the spectral radiance emitted when the red channel is at its maximum intensity
D_R = the digital input to the display
$f_R()$ = a linearization function (discussed further in Section 5.8)

The terms for green and blue are similarly defined. Note that a constant scaling of a spectral radiance function does not change its chromaticity (*x*-*y*) coordinates, hence the term "chromaticity constancy."

These assumptions hold fairly well for many display technologies and result in a simple linear characterization function. Figure 5.7 shows the spectral radiance functions for a typical CRT phosphor set. Sections 5.8 and 5.9 contain more details on CRT and LCD characterization, respectively. Recent research has shown that OLEDs can also be accurately characterized with techniques similar to those described in these sections.[8]

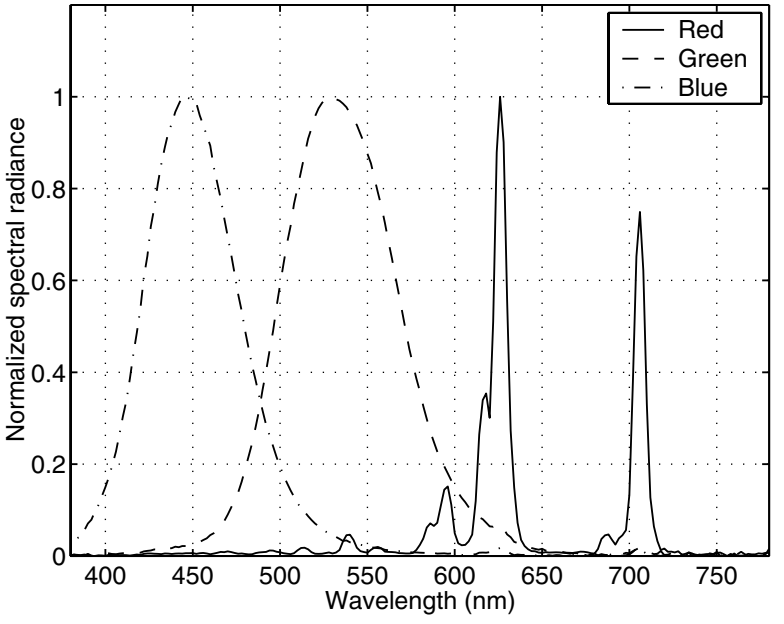


Figure 5.7 Spectral radiance of typical CRT phosphors.

Printing devices produce color via subtractive color mixing in which a base medium for the colorants (usually paper or transparency) reflects or transmits most of the light at all visible wavelengths, and different spectral distributions are produced by combining cyan, magenta, and yellow (CMY) colorants to selectively remove energy from the red, green, and blue portions of the electromagnetic spectrum of a light source. Often, a black colorant (K) is used both to increase the capability to produce dark colors and to reduce the use of expensive color inks. Photographic prints and transparencies and offset, laser, and inkjet printing use subtractive color.

Printers can be broadly classified as being continuous-tone or halftone devices. A continuous-tone process generates uniform colorant layers and modulates the concentration of each colorant to produce different intensity levels. A halftone process generates dots at a small fixed number of concentration levels and modulates the size, shape, and frequency of the dots to produces different intensity levels. (Color halftoning is covered in detail in another chapter.) Both types of processes exhibit complex nonlinear color characteristics, making them more challenging to model and characterize. For one thing, the spectral absorption characteristics of printed colorants do not fulfill the ideal "block dye" assumption, which states that the C, M, Y colorants absorb light in nonoverlapping bands in the long, medium, and short wavelengths, respectively. Such an ideal behavior would result in a simple linear characterization function. Instead, in reality, each of these colorants exhibits unwanted absorptions in other bands, as shown in Figure 5.8, giving rise to complex intercolorant interactions and nonlinear characterization functions. Halftoning introduces additional optical and spatial interactions and thus lends complexity to the characterization function. Nevertheless, much effort has been devoted toward the modeling of continuous and halftone printers as well as toward empirical techniques. A few of these techniques will be explored in further detail in Section 5.10.

A generic workflow for output device calibration and characterization is given in Figure 5.9. A digital target of color patches with known device values is sent to the device. The resulting displayed or printed colors are measured in device-independent (or colorimetric) color coordinates, and a relationship is established between device-dependent and device-

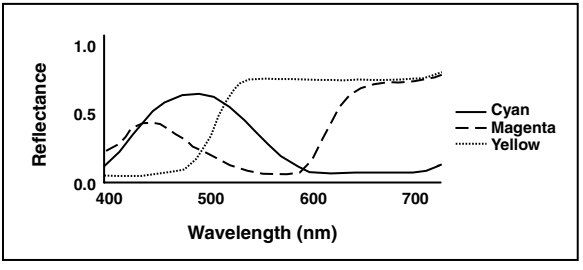


Figure 5.8 Spectral absorption functions of typical C, M, Y colorants.

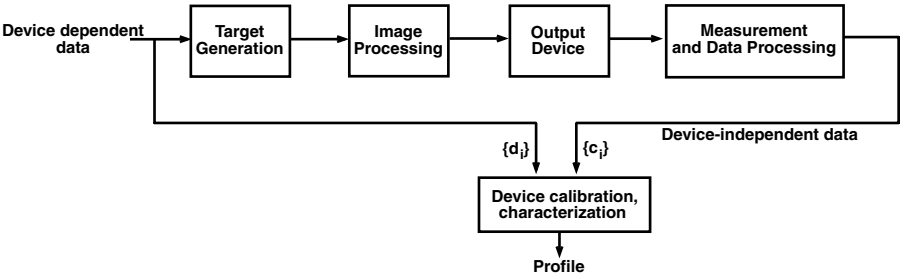


Figure 5.9 Output device characterization workflow.

independent color representations. This can be used to generate both calibration and characterization functions, in that order. For characterization, we once again derive a forward and an inverse function. The forward function describes the colorimetric response of the (calibrated) device to a certain device-dependent input. The inverse characterization function determines the device-dependent values that should be presented to a (calibrated) device to reproduce a certain colorimetric input.

As with input devices, the calibration and characterization should then be evaluated with an independent test target. The flow diagram for doing this is shown in Figure 5.10**.** The test target comprises a set of patches with known

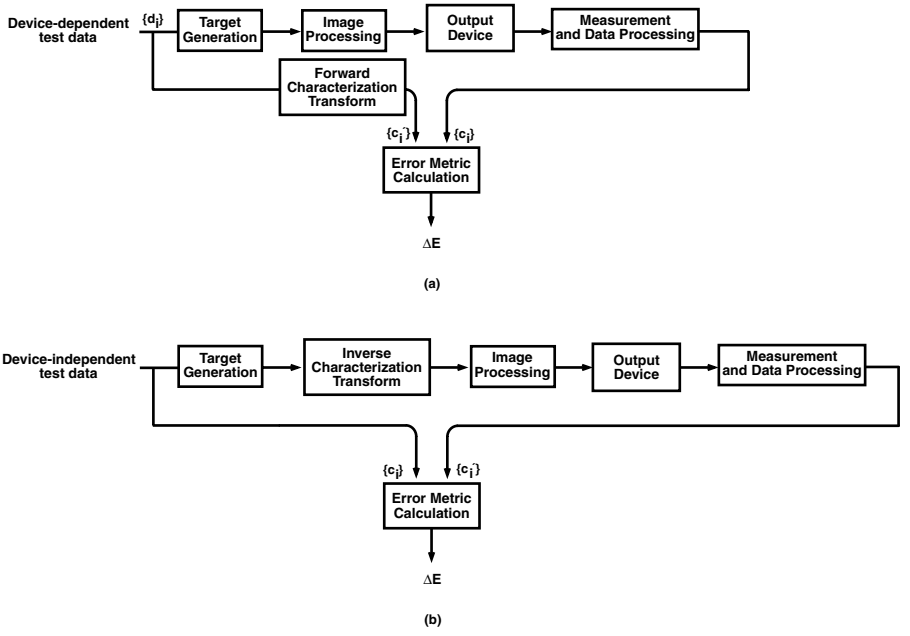


Figure 5.10 Testing of (a) forward and (b) inverse output device characterization.

device-independent coordinates. If calibration is being tested, this target is processed through the calibration functions and rendered to the device. If characterization is being evaluated, the target is processed through both the characterization and calibration function and rendered to the device. The resulting output is measured in device-independent coordinates and compared with the original target values. Once again, the comparison is to be carried out with an appropriate color difference formula such as ΔE^*_{ab} or ΔE^*_{94}.

An important component of the color characteristics of an output device is its color gamut, namely the volume of colors in three-dimensional colorimetric space that is physically achievable by the device. Of particular importance is the gamut surface, as this is used in gamut mapping algorithms. This information can easily be derived from the characterization process. Details of gamut surface calculation are provided in the chapter on gamut mapping.

5.3 Characterization targets and measurement techniques

The generation and measurement of color targets is an important component of device characterization. Hence, a separate section is devoted to this topic.

5.3.1 Color target design

The design of a color target involves several factors. First is the set of colorants and underlying medium of the target. In the case of input devices, the characterization target is created offline (i.e., it is not part of the characterization process) with colorants and media that are representative of what the device is likely to capture. For example, for scanner characterization, photographic and offset lithographic processes are commonly used to create targets on reflective or transmissive media. In the case of output devices, target generation is part of the characterization process and should be carried out using the same colorants and media that will be used for final color rendition.

The second factor is the choice of color patches. Typically, the patches are chosen to span the desired range of the colors to be captured (in the case of input devices) or rendered (in the case of output devices). Often, critical memory colors are included, such as flesh tones and neutrals. The optimal choice of patches is logically a function of the particular algorithm or model that will be used to generate the calibration or characterization function. Nevertheless, a few targets have been adopted as industry standards, and they accommodate a variety of characterization techniques. For input device characterization, these include the CGATS/ANSI IT8.7/1 and IT8.7/2 targets for transmission and reflection media respectively (http://webstore.ansi.org/ansidocstore); the Kodak photographic Q60 target, which is based on the IT8 standards and is made with Ektachrome dyes on Ektacolor paper (www.kodak.com); the GretagMacbeth ColorChecker chart (www.munsell.com); and ColorChecker DC version for digital cam-

eras (www.gretagmacbeth.com). For output device characterization, the common standard is the IT8.7/3 CMYK target (http://webstore.ansi.org/ansidocstore). The Q60 and IT8.7/3 targets are shown in Plates 5A and 5B.

A third factor is the spatial layout of the patches. If a device is known to exhibit spatial nonuniformity, it may be desirable to generate targets with the same set of color patches but rendered in different spatial layouts. The measurements from the multiple targets are then averaged to reduce the effect of the nonuniformity. In general, this approach is advised so as to reduce the overall effect of various imperfections and noise in the characterization process. In the case of input devices, target creation is often not within the practitioner's control; rather, the targets are supplied by a third-party vendor such as Eastman Kodak or Fuji Film. Generally, however, these vendors do use similar principles to generate reliable measurement data.

Another motivation for a specific spatial layout is visual inspection of the target. The Kodak Q60 target, for example, is designed with a gray ramp at the bottom and neutral colors all collected in one area. This allows for convenient visual inspection of these colors, to which we are more sensitive.

5.3.2 Color measurement techniques

5.3.2.1 Visual approaches

Most visual approaches rely on observers making color matching judgments. Typically, a varying stimulus produced by a given device is compared against a reference stimulus of known measurement. When a visual match is reported, this effectively provides a measurement for the varying stimulus and can be correlated with the device value that produced the stimulus. The major advantage of a visual approach is that it does not require expensive measurement instrumentation. Proponents also argue that the best color measurement device is the human visual system, because, after all, this is the basis for colorimetry. However, these approaches have their limitations. First, to achieve reliable results, the visual task must be easy to execute. This imposes severe limits on the number and nature of measurements that can be made. Second, observer-to-observer variation will produce measurements and a characterization that may not be satisfactory to all observers. Nevertheless, visual techniques are appealing in cases where the characterization can be described by a simple model and thus derived with a few simple measurements. The most common application of visual approaches is thus found in CRT characterization, discussed further in Section 5.8.3.

5.3.2.2 Instrument-based approaches

Color measurement instruments fall into two general categories, broadband and narrowband. A broadband measurement instrument reports up to three color signals obtained by optically processing the input light through broadband filters. Photometers are the simplest example, providing a measurement only of the luminance of a stimulus. Their primary use is in determin-

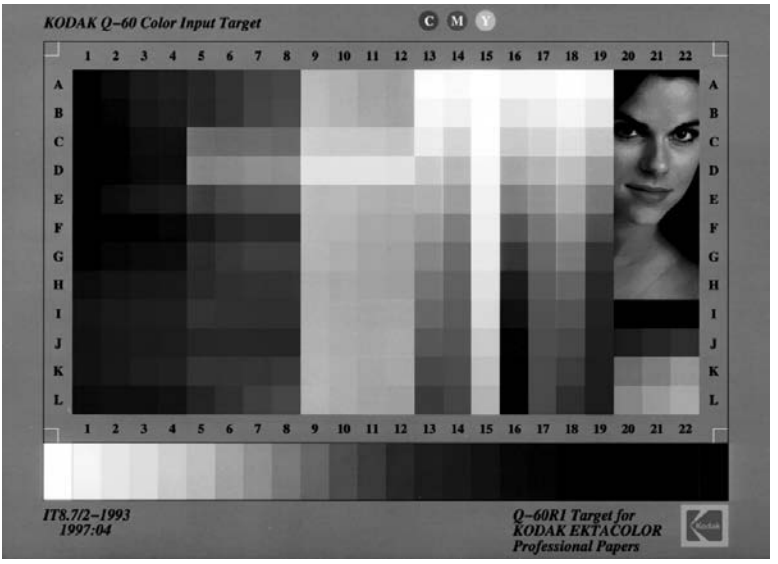


Figure 5A ***(See color insert following page 430)*** Q60 input characterization target.

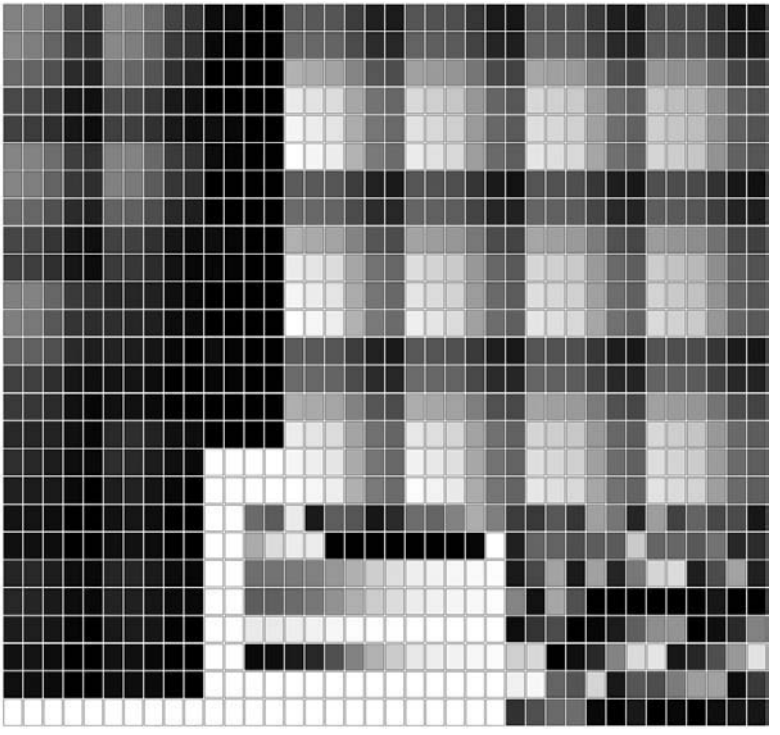


Figure 5B ***(See color insert)*** IT87/3 output characterization target.

ing the nonlinear calibration function of displays (discussed in Section 5.8). Densitometers are an example of broadband instruments that measure optical density of light filtered through red, green, and blue filters. Colorimeters are another example of broadband instruments that directly report tristimulus (XYZ) values and their derivatives such as CIELAB. In the narrowband category fall instruments that report spectral data of dimensionality significantly larger than three. Spectrophotometers and spectroradiometers are examples of narrowband instruments. These instruments typically record spectral reflectance and radiance, respectively, within the visible spectrum in increments ranging from 1 to 10 nm, resulting in 30 to 300 channels. They also have the ability to internally calculate and report tristimulus coordinates from the narrowband spectral data. Spectroradiometers can measure both emissive and reflective stimuli, while spectrophotometers can measure only reflective stimuli.

The main advantages of broadband instruments such as densitometers and colorimeters are that they are inexpensive and can read out data at very high rates. However, the resulting measurement is only an approximation of the true tristimulus signal, and the quality of this approximation varies widely, depending on the nature of the stimulus being measured. Accurate colorimetric measurement of arbitrary stimuli under arbitrary illumination and viewing conditions requires spectral measurements afforded by the more expensive narrowband instruments. Traditionally, the printing industry has satisfactorily relied on densitometers to make color measurements of prints made by offset ink. However, given the larger variety of colorants, printing technologies, and viewing conditions likely to be encountered in today's digital color imaging business, the use of spectral measurement instruments is strongly recommended for device characterization. Fortunately, the steadily declining cost of spectral instrumentation makes this a realistic prospect.

Instruments measuring reflective or transmissive samples possess an internal light source that illuminates the sample. Common choices for sources are tungsten-halogen bulbs as well as xenon and pulsed-xenon sources. An important consideration in reflective color measurement is the optical geometry used to illuminate the sample and capture the reflected light. A common choice is the 45/0 geometry, shown in Figure 5.11. (The two numbers are the angles with respect to the surface normal of the incident illumination and detector respectively.) This geometry is intended to minimize the effect of specular reflection and is also fairly representative of the conditions under which reflection prints are viewed. Another consideration is the measurement aperture, typically set between 3 and 5 mm. Another feature, usually offered at extra cost with the spectrophotometer, is a filter that blocks out ultraviolet (UV) light emanated by the internal source. The filter serves to reduce the amount of fluorescence in the prints that is caused by the UV light. Before using such a filter, however, it must be remembered that common viewing environments are illuminated by light sources (e.g., sunlight, fluorescent lamps) that also exhibit a significant amount of UV

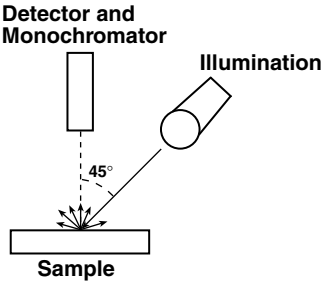


Figure 5.11 45/0 measurement geometry.

energy. Hence, blocking out UV energy may provide color measurements that are less germane to realistic viewing conditions.

For reflective targets, another important factor to consider is the color of the backing surface on which the target is placed for measurement. The two common options are black and white backing, both of which have advantages and disadvantages. A black backing will reduce the effect of show-through from the image on the backside of a duplex print. However, it will also expose variations in substrate transmittance, thus resulting in noisier measurements. A white backing, on the other hand, is not as effective at attenuating show-through; however, the resulting measurements are less noisy, because the effect of substrate variations is reduced. Generally, a white backing is recommended if the target is not duplex (which is typically the case.) Further details are provided by Rich.[9]

Color measurement instruments must themselves be calibrated to output reliable and repeatable data. Instrument calibration entails understanding and specifying many of the aforementioned parameters and, in some cases, needs to be carried out frequently. Details are provided by Zwinkel.[10]

Because color measurement can be a labor-intensive task, much has been done in the color management industry to automate this process. The Gretag Spectrolino™ product enables the target to be placed on a stage and automatically measured by the instrument. These measurements are then stored on a computer to be retrieved for deriving the characterization. In a similar vein, X-Rite Corporation has developed the DTP-41 scanning spectrophotometer. The target is placed within a slot in the "strip reader" and is automatically moved through the device as color measurements are made of each patch.

5.3.3 *Absolute and relative colorimetry*

An important concept that underlies device calibration and characterization is normalization of the measurement data by a reference white point. Recall from an earlier chapter that the computation of tristimulus XYZ values from spectral radiance data is given by

$$X = K \int_{\lambda \varepsilon V} S(\lambda)\bar{x}(\lambda)\partial\lambda, \quad Y = K \int_{\lambda \varepsilon V} S(\lambda)\bar{y}\partial\lambda, \quad Z = K \int_{\lambda \varepsilon V} S(\lambda)\bar{z}(\lambda)\partial\lambda \qquad (5.9)$$

where $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ = color matching functions
V = set of visible wavelengths
K = a normalization constant

In absolute colorimetry, K is a constant, expressed in terms of the maximum efficacy of radiant power, equal to 683 lumens/W. In relative colorimetry, K is chosen such that $Y = 100$ for a chosen reference white point.

$$K = \frac{100}{\int_{\lambda \varepsilon V} S_w(\lambda)\bar{y}(\lambda)\partial\lambda} \qquad (5.10)$$

where $S_w(\lambda)$ = the spectral radiance of the reference white stimulus.

For reflective stimuli, radiance $S_w(\lambda)$ is a product of incident illumination $I(\lambda)$ and spectral reflectance $R_w(\lambda)$ of a white sample. The latter is usually chosen to be a perfect diffuse reflector (i.e., $R_w(\lambda) = 1$) so that $S_w(\lambda) = I(\lambda)$ in Equation 5.10.

There is an additional white-point normalization to be considered. The conversion from tristimulus values to appearance coordinates such as CIELAB or CIELUV requires the measurement of a reference white stimulus and an appropriate scaling of all tristimulus values by this white point. In the case of emissive display devices, the white point is the measurement of the light emanated by the display device when the driving RGB signals are at their maximal values (e.g., $D_R = D_G = D_B = 255$ for 8-bit input). In the case of reflective samples, the white point is obtained by measuring the light emanating from a reference white sample illuminated by a specified light source. If an ideal diffuse reflector is used as the white sample, we refer to the measurements as being in *media absolute colorimetric coordinates.* If a particular medium (e.g., paper) is used as the stimulus, we refer to the measurements as being in *media relative colorimetric coordinates.* Conversions between media absolute and relative colorimetry are achieved with a white-point normalization model such as the von Kries formula.

To get an intuitive understanding of the effect of media absolute vs. relative colorimetry, consider an example of scan-to-print reproduction of a color image. Suppose the image being scanned is a photograph whose medium typically exhibits a yellowish cast. This image is to be printed on a xerographic printer, which typically uses a paper with fluorescent whiteners and is thus lighter and bluer than the photographic medium. The image is scanned, processed through both scanner and printer characterization functions, and printed. If the characterizations are built using media absolute colorimetry, the yellowish cast of the photographic medium is

preserved in the xerographic reproduction. On the other hand, with media relative colorimetry, the "yellowish white" of the photographic medium maps directly to the "bluish white" of the xerographic medium under the premise that the human visual system adapts and perceives each medium as "white" when viewed in isolation. Arguments can be made for both modes, depending on the application. Side-by-side comparisons of original and reproduction may call for media absolute characterization. If the reproduction is to be viewed in isolation, it is probably preferable to exploit visual white-point adaptation and employ relative colorimetry. To this end, the ICC specification supports both media absolute and media relative modes in its characterization tables.

Finally, we remark that, while a wide variety of standard illuminants can be selected for deriving the device characterization function, the most common choices are CIE daylight illuminants D5000 (typically used for reflection prints) and D6500 (typically used for the white point of displays).

5.4 *Multidimensional data fitting and interpolation*

Another critical component underlying device characterization is multidimensional data fitting and interpolation. This topic is treated in general mathematical terms in this section. Application to specific devices will be discussed in ensuing sections.

Generally, the data samples generated by the characterization process in both device-dependent and device-independent spaces will constitute only a small subset of all possible digital values that could be encountered in either space. One reason for this is that the total number of possible samples in a color space is usually prohibitively large for direct measurement of the characterization function. As an example, for R, G, B signals represented with 8-bit precision, the total number of possible colors is 2^{24} = 16,777,216; clearly an unreasonable amount of data to be acquired manually. However, because the final characterization function will be used for transforming arbitrary image data, it needs to be defined for all possible inputs within some expected domain. To accomplish this, some form of data fitting or interpolation must be performed on the characterization samples. In model-based characterization, the underlying physical model serves to perform the fitting or interpolation for the forward characterization function. With empirical approaches, mathematical techniques may be used to perform data fitting or interpolation. Some of the common mathematical approaches are discussed in this section.

The fitting or interpolation concept can be formalized as follows. Define a set of T m-dimensional device-dependent color samples $\{\mathbf{d}_i\} \in R^m$, $i = 1, \ldots, T$ generated by the characterization process. Define the corresponding set of n-dimensional device-independent samples $\{\mathbf{c}_i\} \in R^n$, $i = 1, \ldots, T$. For the majority of characterization functions, $n = 3$, and $m = 3$ or 4. We will often refer to the pair $(\{\mathbf{d}_i\}, \{\mathbf{c}_i\})$ as the set of training samples. From this set, we wish to evaluate one or both of the following functions:

- f: $\boldsymbol{F} \in R^m \rightarrow R^n$, mapping device-dependent data within a domain $\boldsymbol{F}$ to device-independent color space
- g: $\boldsymbol{G} \in R^n \rightarrow R^m$, mapping device-independent data within a domain $\boldsymbol{G}$ to device-dependent color space

In interpolation schemes, the error of the functional approximation is identically zero at all the training samples, i.e., $f(\mathbf{d}_i) = \mathbf{c}_i$, and $g(\mathbf{c}_i) = \mathbf{d}_i$, $i = 1, \ldots, T$.

In fitting schemes, this condition need not hold. Rather, the fitting function is designed to minimize an error criterion between the training samples and the functional approximations at these samples. Formally,

$$f_{opt} = \arg\min_{f} E_1(|\mathbf{c}_i, f(\mathbf{d}_i)|_{i=1,\ldots,T}); \quad g_{opt} = \arg\min_{g} E_2(|\mathbf{d}_i, g(\mathbf{c}_i)|_{i=1,\ldots,T}) \tag{5.11}$$

where E_1 and E_2 are suitably chosen error criteria.

A common approach is to pick a parametric form for f (or g) and minimize the mean squared error metric, given by

$$E_1 = \frac{1}{T}\sum_{i=1}^{T} \| \boldsymbol{c}_i - f(\mathbf{d}_i) \|^2 \tag{5.12}$$

An analogous expression holds for E_2. The minimization is performed with respect to the parameters of the function f or g.

Unfortunately, most of the data fitting and interpolation approaches to be discussed shortly are too computationally expensive for the processing of large amounts of image pixel data in real time. The most common way to address this problem is to first evaluate the complex fitting or interpolation functions at a regular lattice of points in the input space and build a multidimensional lookup table (LUT). A fast interpolation technique such as trilinear or tetrahedral interpolation is then used to transform image data using this LUT. The subject of fast LUT interpolation on regular lattices is treated in a later chapter. Here, we will focus on the fitting and interpolation methods used to initially approximate the characterization function and build the LUT.

Often, it is necessary to evaluate the functions f and g within domains **F** and **G** that are outside of the volumes spanned by the training data $\{\mathbf{d}_i\}$ and $\{\mathbf{c}_i\}$. An example is shown in Figure 5.12 for printer characterization mapping CIELAB to CMY. A two-dimensional projection of CIELAB space is shown, with a set of training samples $\{\mathbf{c}_i\}$ indicated by "x." Device-dependent CMY values $\{\mathbf{d}_i\}$ are known at each of these points. The shaded area enclosed by these samples is the range of colors achievable by the printer, namely its color gamut. From these data, the inverse printer characterization function from CIELAB to CMY is to be evaluated at each of the lattice points lying on the three-dimensional lookup table grid (projected as a two-dimensional grid in

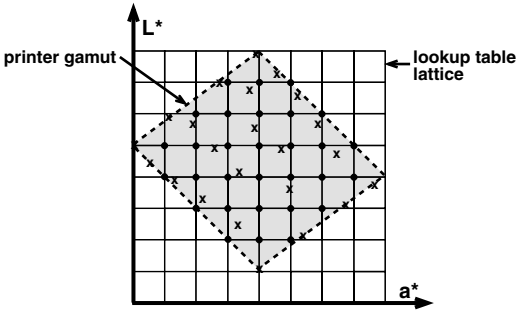


Figure 5.12 Multidimensional lattice in CIELAB, overlaying printer gamut.

Figure 5.12). Hence, the domain **G** in this case is the entire CIELAB cube. Observe that a fraction of these lattice points lie within the printer gamut (shown as black circles). Interpolation or data fitting of these points is usually well defined and mathematically robust, since a sufficient amount of training data is available in the vicinity of each lattice point. However, a substantial fraction of lattice points also lie outside the gamut, and there are no training samples in the vicinity of these points. One of two approaches can be used to determine the characterization function at these points.

1. Apply a preprocessing step that first maps all out-of-gamut colors to the gamut, then perform data fitting or interpolation to estimate output values.
2. Extrapolate the fitting or interpolation function to these out-of-gamut regions.

Some of the techniques described herewith allow for data extrapolation. The latter will invariably generate output data that lie outside the allowable range in the output space. Hence, some additional processing is needed to limit the data to this range. Often, a hard-limiting or clipping function is employed to each of the components of the output data.

Two additional comments are noteworthy. First, while the techniques described in this section focus on fitting and interpolation of multidimensional data, most of them apply in a straightforward manner to one-dimensional data typically encountered in device calibration. Linear and polynomial regression and splines are especially popular choices for fitting one-dimensional data. Lattice-based interpolation reduces trivially to piecewise linear interpolation, and it can be used when the data are well behaved and exhibit low noise. Secondly, the reader is strongly encouraged, where possible, to plot the raw data along with the fitting or interpolation function to obtain insight on both the characteristics of the data and the functional approximation. Often, data fitting involves a delicate balance between accurately approximating the function and smoothing out the noise. This balance

is difficult to achieve by examining only a single numerical error metric and is significantly aided by visualizing the entire dataset in combination with the fitting functions.

5.4.1 Linear least-squares regression

This very common data fitting approach is used widely in color imaging, particularly in device characterization and modeling. The problem is formulated as follows. Denote **d** and **c** to be the input and output color vectors, respectively, for a characterization function. Specifically, **d** is a $1 \times m$ vector, and **c** is a $1 \times n$ vector. We wish to approximate the characterization function by the linear relationship $\mathbf{c} = \mathbf{d} \cdot \mathbf{A}$.

The matrix **A** is of dimension $m \times n$ and is derived by minimizing the mean squared error of the linear fit to a set of training samples, $\{\mathbf{d}_i, \mathbf{c}_i\}$, $i = 1, \ldots, T$. Mathematically, the optimal **A** is given by

$$A_{opt} = \arg\min_{A} \left\{ \frac{1}{T} \sum_{i=1}^{T} \| \boldsymbol{c}_i - \mathbf{d}_i \mathbf{A} \|^2 \right\} \tag{5.13}$$

To continue the formulation, it is convenient to collect the samples $\{\mathbf{c}_i\}$ into a $T \times n$ matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_T]$, and $\{\mathbf{d}_i\}$ into a $T \times m$ matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_T]$. The linear relationship is given by $\mathbf{C} = \mathbf{D} \cdot \mathbf{A}$. The optimal **A** is given by $\mathbf{A} = \mathbf{D}\dagger\ \mathbf{C}$, where $\mathbf{D}\dagger$ is the generalized inverse (sometimes known as the Moore–Penrose pseudo-inverse) of **D**. In the case where $\mathbf{D}^t\mathbf{D}$ is invertible, the optimum **A** is given by

$$\mathbf{A} = (\mathbf{D}^t\mathbf{D})^{-1}\ \mathbf{D}^t\mathbf{C} \tag{5.14}$$

See Appendix 5.A for the derivation and numerical computation of this least-squares solution. It is important to understand the conditions for which the solution to Equation 5.14 exists. If $T < m$, we have an underdetermined system of equations with no unique solution. The mathematical consequence of this is that the matrix $\mathbf{D}^t\mathbf{D}$ is of insufficient rank and is thus not invertible. Thus, we need at least as many samples as the dimensionality of the input data. If $T = m$, we have an exact solution for **A** that results in the squared error metric being identically zero. If $T > m$ (the preferred case), Equation 5.14 provides a least-squares solution to an overdetermined system of equations. Note that linear regression affords a natural means of extrapolation for input data **d** lying outside the domain of the training samples. As mentioned earlier, some form of clipping will be needed to limit such extrapolated outputs to their allowable range.

5.4.2 Weighted least-squares regression

The standard least-squares regression can be extended to minimize a weighted error criterion,

$$\boldsymbol{A}_{opt} = \arg\min\left\{\frac{1}{T}\sum_{i=1}^{T} w_i \|\mathbf{c}_i - \mathbf{d}_i\mathbf{A}\|^2\right\} \tag{5.15}$$

where w_i = positive-valued weights that indicate the relative importance of the ith data point, $\{\mathbf{d}_i, \mathbf{c}_i\}$.

Adopting the notation in Section 5.4.1, a straightforward extension of Appendix 5.A results in the following optimum solution:

$$\mathbf{A} = (\mathbf{D}^t\mathbf{W}\mathbf{D})^{-1}\,\mathbf{D}^t\,\mathbf{W}\mathbf{C} \tag{5.16}$$

where $\mathbf{W}$ is a $T \times T$ diagonal matrix with diagonal entries w_i.

The resulting fit will be biased toward achieving greater accuracy at the more heavily weighted samples. This can be a useful feature in device characterization when, for example, we wish to assign greater importance to colors in certain regions of color space (e.g., neutrals, fleshtones, etc.). As another example, in spectral regression, it may be desirable to assign greater importance to certain wavelengths than others.

5.4.3 Polynomial regression

This is a special form of least-squares fitting wherein the characterization function is approximated by a polynomial. We will describe the formulation using, as an example, a scanner characterization mapping device RGB space to XYZ tristimulus space. The formulation is conceptually identical for input and output devices and for the forward and inverse functions.

The third-order polynomial approximation for a transformation from RGB to XYZ space is given by

$$X = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} w_{X,l} R^i G^j B^k; \quad Y = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} w_{Y,l} R^i G^j B^k;$$

$$Z = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} w_{Z,l}\; R^i G^j B^k \tag{5.17}$$

where $w_{X,l}$, etc. = polynomial weights

l = a unique index for each combination of i, j, k

In practice, several of the terms in Equation 5.17 are eliminated (i.e., the weights w are set to zero) so as to control the number of degrees of freedom in the polynomial. Two common examples, a linear and third-order approximation,

are given below. For brevity, only the X term is defined; analogous definitions hold for Y and Z.

$$X = w_{X,0}R + w_{X,1}G + w_{X,2}B \tag{5.18a}$$

$$\begin{aligned} X = {} & w_{X,0} + w_{X,1}R + w_{X,2}G + w_{X,3}B + w_{X,4}RG + w_{X,5}GB \\ & + w_{X,6}RB + w_{X,7}R^2 + w_{X,8}G^2 + w_{X,9}B^2 + w_{X,10}RGB \end{aligned} \tag{5.18b}$$

In matrix-vector notation, Equation 5.17 can be written as

$$\begin{bmatrix} X & Y & Z \end{bmatrix} = \begin{bmatrix} 1 & R & G & \ldots & R^3 & G^3 & B^3 \end{bmatrix} \begin{bmatrix} w_{X,0} & w_{Y,0} & w_{Z,0} \\ w_{X,1} & w_{Y,1} & w_{z,3} \\ & \cdots & \\ w_{X,63} & w_{Y,63} & w_{Z,63} \end{bmatrix} \tag{5.19}$$

or more compactly,

$$\mathbf{c} = \mathbf{p} \cdot \mathbf{A} \tag{5.20}$$

where $\mathbf{c}$ = output XYZ vector
$\mathbf{p}$ = $1 \times Q$ vector of Q polynomial terms derived from the input RGB vector $\mathbf{d}$
$\mathbf{A}$ = $Q \times 3$ matrix of polynomial weights to be optimized

In the complete form, $Q = 64$. However, with the more common simplified approximations in Equation 5.18, this number is significantly smaller; i.e., $Q = 3$ and $Q = 11$, respectively.

Note from Equation 5.20 that the polynomial regression problem has been cast into a linear least-squares problem with suitable preprocessing of the input data $\mathbf{d}$ into the polynomial vector $\mathbf{p}$. The optimal $\mathbf{A}$ is now given by

$$\mathbf{A}_{opt} = \arg\min_{\mathrm{A}} \left\{ \frac{1}{T} \sum_{i=1}^{T} \left\| \mathbf{c}_i - \mathbf{p}_i \mathbf{A} \right\|^2 \right\} \tag{5.21}$$

Collecting the samples $\{\mathbf{c}_i\}$ into a $T \times 3$ matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_T]$, and $\{\mathbf{p}_i\}$ into a $T \times Q$ matrix $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_T]$, we have the relationship $\mathbf{C} = \mathbf{P} \cdot \mathbf{A}$. Following the formulation in Section 5.4.1, the optimal solution for $\mathbf{A}$ is given by

$$\mathbf{A} = (\mathbf{P}^t\mathbf{P})^{-1}\,\mathbf{P}^t\mathbf{C} \tag{5.22}$$

For the $Q \times Q$ matrix $(\mathbf{P}^t\mathbf{P})$ to be invertible, we now require that $T \geq Q$.

Polynomial regression can be summarized as follows:

1. Select a set of T training samples, where $T > Q$, the number of terms in the polynomial approximation. It is recommended that the samples adequately span the input color space.
2. Use the assumed polynomial model to generate the polynomial terms $\mathbf{p}_i$ from the input data $\mathbf{d}_i$. Collect $\mathbf{c}_i$ and $\mathbf{p}_i$ into matrices $\mathbf{C}$ and $\mathbf{P}$, respectively.
3. Use Equation (5.22 to derive the optimal $\mathbf{A}$.
4. For a given input color $\mathbf{d}$, use the same polynomial model to generate the polynomial terms $\mathbf{p}$.
5. Use Equation 5.20 to compute the output color $\mathbf{c}$.

Figure 5.13 is a graphical one-dimensional example of different polynomial approximations to a set of training samples. The straight line is a linear fit (Q = 3) and is clearly inadequate for the given data. The solid curve is a second-order polynomial function (Q = 7) and offers a much superior fit. The dash–dot curve closely following the solid curve is a third-order polynomial approximation (Q = 11). Clearly, this offers no significant advantage over the second-order polynomial. In general, we recommend using the smallest number of polynomial terms that adequately fits the curvature of the function while still smoothing out the noise. This choice is dependent on the particular device characteristics and is obtained by experimentation, intuition, and experience. Finally, it is noted that polynomial regression affords a natural means of extrapolation for input data lying outside the domain of the training samples.

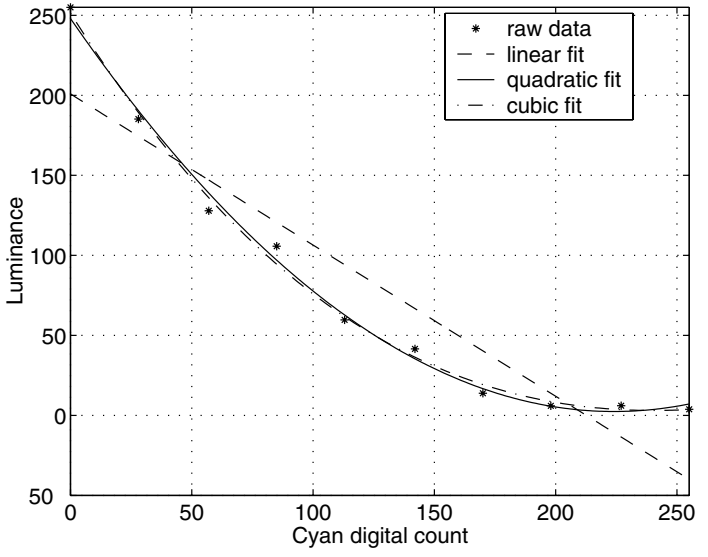


Figure 5.13 One-dimensional example of different polynomial approximations.

5.4.4 Distance-weighted techniques

The previous section described the use of a global polynomial function that results in the best overall fit to the training samples. In this section, we describe a class of techniques that also employ simple parametric functions; however, the parameters vary across color space to best fit the local characteristics of the training samples.

5.4.4.1 Shepard's interpolation

This is a technique that can be applied to cases in which the input and output spaces of the characterization function are of the same dimensionality. First, a crude approximation of the characterization function is defined: $\hat{c} = f_{approx}(\mathbf{d})$. The main purpose of $f_{approx}()$ is to bring the input data into the orientation of the output color space. (By "orientation," it is meant that all RGB spaces are of the same orientation, as are all luminance–chrominance spaces, etc.) If both color spaces are already of the same orientation, e.g., printer RGB and sRGB, we can simply let $f_{approx}()$ be an identity function so that $\hat{c} = \mathbf{d}$. If, for example, the input and output spaces are scanner RGB and CIELAB, an analytic transformation from any colorimetric RGB (e.g., sRGB) to CIELAB could serve as the crude approximation.

Next, given the training samples $\{\mathbf{d}_i\}$ and $\{\mathbf{c}_i\}$ in the input and output space, respectively, we define error vectors between the crude approximation and true output values of these samples: $e_i = c_i - \hat{c}_i, = 1, \ldots, T$. Shepard's interpolation for an arbitrary input color vector $\mathbf{d}$ is then given by[11]

$$\mathbf{c} = \hat{\mathbf{c}} + K_w \sum_{i=1}^{T} w(\mathbf{d} - \mathbf{d}_i)\mathbf{e}_i \tag{5.23}$$

where $w()$ = weights

K_w = a normalizing factor that ensures that these weights sum to unity as follows:

$$K_w = \frac{1}{\sum_{i=1}^{T} w(\mathbf{d} - \mathbf{d}_i)} \tag{5.24}$$

The second term in Equation 5.23 is a correction for the residual error between $\mathbf{c}$ and $\hat{\mathrm{c}}$, and it is given by a weighted average of the error vectors $\mathbf{e}_i$ at the training samples. The weighting function $w()$ is chosen to be inversely proportional to the Euclidean distance between $\mathbf{d}$ and $\mathbf{d}_i$ so that training samples that are nearer the input point exhibit a stronger influence than those that are further away. There are numerous candidates for $w()$. One form that has been successfully used for printer and scanner characterization is given by[12]

$$w(\mathbf{d} - \mathbf{d}_i) = \frac{1}{\|\mathbf{d} - \mathbf{d}_i\|^p + \varepsilon} \tag{5.25}$$

where $\|\mathbf{d} - \mathbf{d}_i\|$ denotes Euclidean distance between vectors $\mathbf{d}$ and $\mathbf{d}_i$, and ρ and ε are parameters that dictate the relative influence of the training samples as a function of their distance from the input point. As ρ increases, the influence of a training sample decays more rapidly as a function of its distance from the input point. As ε increases, the weights become less sensitive to distance, and the approach migrates from a local to a global approximation.

Note that, in the special case where ε = 0, the function in Equation 5.25 has a singularity at $\mathbf{d} = \mathbf{d}_i$. This can be accommodated by adding a special condition to Equation 5.23.

$$\mathbf{c} = \begin{cases} \hat{\mathbf{c}} + K_w \sum_{i=1}^{T} w(\mathbf{d} - \mathbf{d}_i)\mathbf{e}_i & if(\|\mathbf{d} - \mathbf{d}_i\| \geq t) \\ \mathbf{c}_i & if\|\mathbf{d} - \mathbf{d}_i\| < t \end{cases} \tag{5.26}$$

where $w()$ = given by Equation 5.25 with ε = 0
t = a suitably chosen distance threshold that avoids the singularity at $\mathbf{d} = \mathbf{d}_i$

Other choices of $w()$ include the Gaussian and exponential functions.[11] Note that, depending on how the weights are chosen, Shepard's algorithm can be used for both data fitting (i.e., Equation 5.23 and Equation 5.25 with ε > 0), and data interpolation, wherein the characterization function coincides exactly at the training samples (i.e., Equation 5.26). Note also that this technique allows for data extrapolation. As one moves farther away from the volume spanned by the training samples, the distances $\|\mathbf{d} - \mathbf{d}_i\|$ and hence the weights $w()$ approach a constant. In the limit, the overall error correction in Equation 5.23 is an unweighted average of the error vectors $\mathbf{e}_i$.

5.4.4.2 Local linear regression

In this approach, the form of the characterization function that maps input colors $\mathbf{d}$ to output colors $\mathbf{c}$ is given by

$$\mathbf{c} = \mathbf{d} \cdot \mathbf{A}_d \tag{5.27}$$

This looks very similar to the standard linear transformation, the important difference being that the matrix $\mathbf{A}_d$ now varies as a function of the input color $\mathbf{d}$ (hence the term *local linear regression)*. The optimal $\mathbf{A}_d$ is obtained by a distance-weighted least-squares regression,

$$\mathbf{A}_d^{opt} = \arg\min\left\{\frac{1}{T}\sum_{i=1}^{T}\|\mathbf{c}_i - \mathbf{d}_i\mathbf{A}_d\|^2 w(\mathbf{d}-\mathbf{d}_i)\right\} \tag{5.28}$$

As with Shepard's interpolation, the weighting function $w()$ is inversely proportional to the Euclidean distance $\|\boldsymbol{d} - \boldsymbol{d}_i\|$, so training samples $\mathbf{d}_i$ that are farther away from the input point $\mathbf{d}$ are assigned a smaller weight than nearby points. A form such as Equation 5.25 may be used.[12] The solution is given by Equation 5.16 in Section 4.2, where the weights $w(\mathbf{d} - \mathbf{d}_i)$ constitute the diagonal terms of $\mathbf{W}$. Note that because $w()$ is a function of the input vector $\mathbf{d}$, Equation 5.16 must be recalculated for every input vector $\mathbf{d}$. Hence, this is a computationally intensive algorithm. Fortunately, as noted earlier, this type of data fitting is not applied to image pixels in real time. Instead, it is used offline to create a multidimensional lookup table.

Figure 5.14 is a one-dimensional example of the locally linear transform using the inverse-distance weighting function, Equation 5.25. As with Shepard's interpolation, ρ and ε affect the relative influence of the training samples as a function of distance. The plots in Figure 5.14 were generated with ρ = 4 and compare two values of ε. For ε = 0.001, the function closely follows the data. As ε increases to 0.01, the fit averages the fine detail while preserving the gross curvature. In the limit as ε increases, $w()$ in Equation 5.25 approaches a constant, the technique approaches global linear regression, and the fit approaches a straight line. Similar trends hold for ρ. These param-

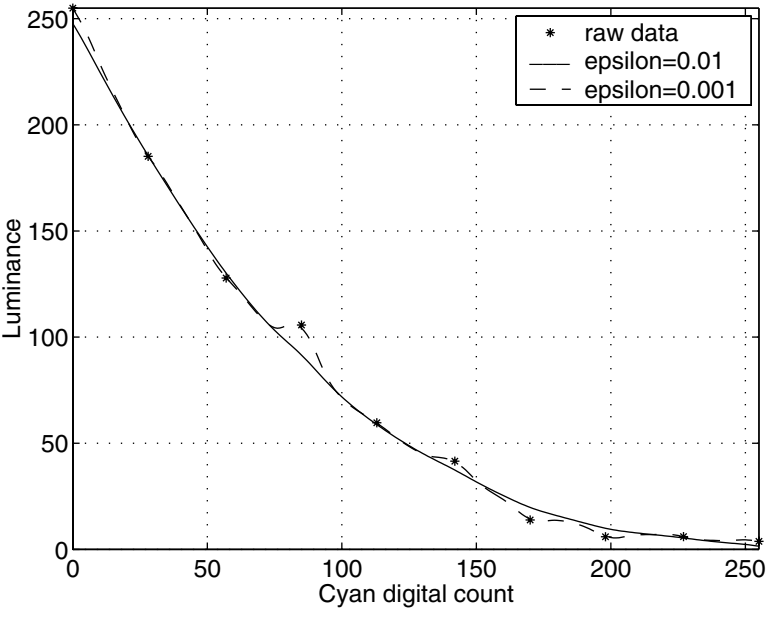


Figure 5.14 Local linear regression for different values of ε.

eters thus offer direct control on the amount of curvature and smoothing that occurs in the data fitting process and should be chosen based on *a priori* knowledge about the device and noise characteristics.

As with Shepard's algorithm, this approach also allows for data extrapolation. As the input point moves farther away from the volume spanned by the training samples, the weights $w()$ approach a constant, and we are again in the regime of global linear extrapolation.

5.4.5 Lattice-based interpolation

In this class of techniques, the training samples are assumed to lie on a regular lattice in either the input or output space of the characterization function. Define l_i to be a set of real-valued levels along the ith color dimension. A regular lattice L^m in m-dimensional color space is defined as the set of all points $\mathbf{x} = [x_1, \ldots, x_m]^t$ whose ith component x_i belongs to the set l_i. Mathematically, the lattice can be expressed as

$$L^m = \{x \in R^m \mid x_i \in l_i, i = 1,\ldots,m\} \text{ or, equivalently, } L^m = \prod_{i=1}^{m} l_i \qquad (5.29)$$

where the second expression is a Cartesian product. If s_i is the number of levels in l_i, the size of the lattice is the product $s_1 \times s_2 \times \ldots \times s_m$. Commonly, all the l_i are identical sets of size s, resulting in a lattice of size s^m.

In one dimension, a lattice is simply a set of levels $\{x_j\}$ in the input space. Associated with these levels are values $\{y_j\}$ in the output space. Evaluation of the one-dimensional function for an intermediate value of x is then performed by finding the interval $[x_j, x_{j+1}]$ that encloses x and performing piecewise interpolation using either linear or nonlinear functions. If sufficient samples exist and exhibit low noise, linear interpolation can be used as follows:

$$y = y_j + \left(\frac{x - x_j}{x_{j+1} - x_j}\right)(y_{j+1} - y_j) \qquad (5.30)$$

If only a sparse sampling is available, nonlinear functions such as splines may be a better choice (see Section 5.4.8).

Let us turn to the more interesting multidimensional case. A three-dimensional lattice in CMY space is shown in Figure 5.15, along with the corresponding lattice in CIELAB space. The lines indicate the levels l_i along each dimension, and the intersections of these lines are the lattice points. The lattice size in this example is $5 \times 5 \times 5 = 125$. A lattice partitions a color space into a set of smaller subvolumes. The characterization transform is executed in two steps: (1) Locate the subvolume to which an input color belongs, and (2) perform some form of interpolation, effectively a distance-weighted average, among the neighboring lattice points. By definition, the characterization function will coincide with the training samples at the lattice points.

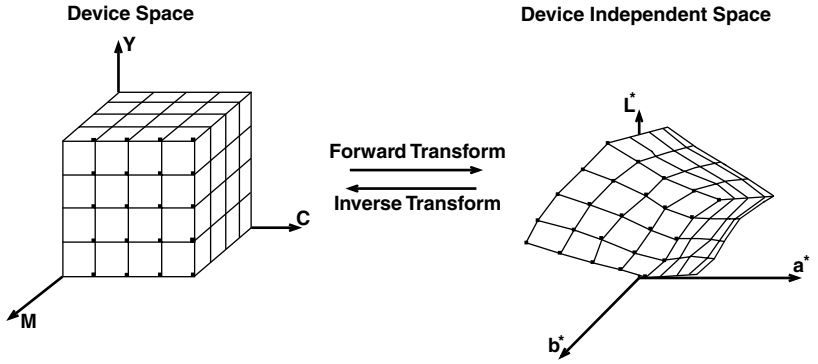


Figure 5.15 Three-dimensional lattice in CMY and CIELAB space.

Note from Figure 5.15 that, while the lattice is regular in one space, it need not be regular in the other space. In the case of the forward characterization function for an output device, the regular lattice exists in the input domain of the function. Efficient interpolation techniques exist for regular lattices, including trilinear, tetrahedral, prism, and pyramidal interpolation. These are described in detail in Chapter 11 and thus will not be discussed here. The more challenging case is evaluation of the inverse transform, whereby the lattice that partitions the input domain of the function is irregular. We will describe a solution to this problem known as tetrahedral inversion.[13] Let us assume that the dimensionality of both input and output color spaces are equal and assume, without less of generality, that the data are three-dimensional. A regular lattice in three-dimensional space provides a partitioning into a set of sub-cubes. Each sub-cube can be further partitioned into several tetrahedra, as shown in Figure 5.16. A tetrahedron is a volume bounded by four vertices and four planar surfaces. There are several ways to split a cube into tetrahedra, the most common form being a partitioning

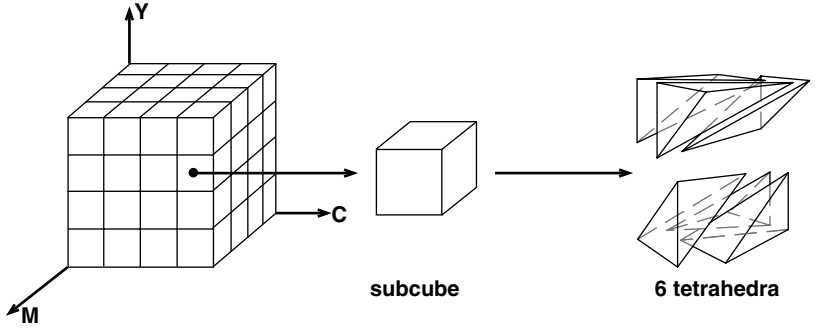


Figure 5.16 Partitioning of color space into cubes, further subdivided into tetrahedra.

into six tetrahedra that share a common diagonal of the cube. An association is now established between each quadruplet of vertices that constitute a tetrahedron on the regular lattice in device space and the corresponding quadruplet of vertices on the irregular lattice in device-independent space, as shown in Figure 5.17. The inverse characterization function $g()$ is then modeled as one that maps each tetrahedral volume in device-independent space to a corresponding tetrahedral volume in device space.

Specifically, referring to Figure 5.17, let $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4\}$ be four vertices of a tetrahedron $\mathbf{T}_d$ in device space, and $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$ be the corresponding vertices forming a tetrahedron $\mathbf{T}_c$ in device-independent space. Here, $\mathbf{d}_i$ and $\mathbf{c}_i$ are 3×1 vectors. Given a point $\mathbf{c}$ lying within $\mathbf{T}_c$, the corresponding point $\mathbf{d}$ in $\mathbf{T}_d$ is given by

$$\mathbf{d} = g(\mathbf{c}) = \mathbf{A}_\mathbf{d} \cdot \mathbf{A}_\mathbf{c}^{-1} \cdot (\mathbf{c} - \mathbf{c}_1) + \mathbf{d}_1 \tag{5.31}$$

where $\mathbf{A}_\mathbf{d}$ and $\mathbf{A}_\mathbf{c}$ are 3×3 matrices given by

$$\mathbf{A}_\mathbf{d} = [\mathbf{d}_2 - \mathbf{d}_1 \;\; \mathbf{d}_3 - \mathbf{d}_1 \;\; \mathbf{d}_4 - \mathbf{d}_1]; \;\; \mathbf{A}_\mathbf{c} = [\mathbf{c}_2 - \mathbf{c}_1 \;\; \mathbf{c}_3 - \mathbf{c}_1 \;\; \mathbf{c}_4 - \mathbf{c}_1] \tag{5.32}$$

Equation 5.31 tells us that $g()$ is being modeled as a piecewise affine function. It can be shown that $\mathbf{c}$ is included within a tetrahedron $\mathbf{T}_c$ if all the elements of the vector $\mathbf{A}_\mathbf{c}^{-1}(\mathbf{c} - \mathbf{c}_1)$ are nonnegative and their sum lies between 0 and 1.[13]

Tetrahedral inversion may be summarized as follows:

- Partition the regular lattice of training samples into a set of tetrahedra.
- Establish a correspondence between tetrahedra on the regular lattice in the one space and tetrahedra on the possibly irregular lattice in the other space.
- Given an input point $\mathbf{c}$, find the tetrahedron $\mathbf{T}_\mathbf{c}$ to which the point belongs, using the aforementioned membership test.

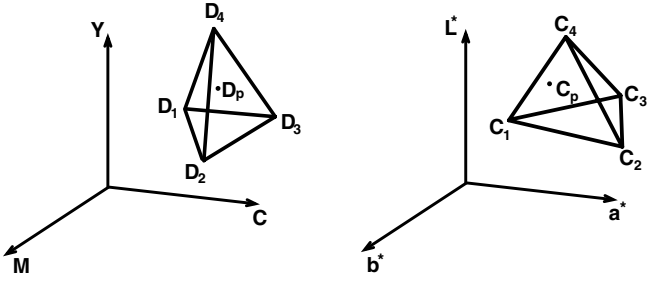


Figure 5.17 Tetrahedral mapping from device CMY space to colorimetric CIELAB space.

- Use Equations 5.31 and 5.32 to evaluate the characterization function $\mathbf{d} = g(\mathbf{c})$.

Because tetrahedral inversion requires membership in a tetrahedron, it does not allow extrapolation to points **c** that lie outside the lattice defined by the training samples. Hence, such points must first be mapped to the lattice volume before carrying out the inversion algorithm. Also, it is worth noting that tetrahedral interpolation on a regular lattice can be implemented with a highly simplified form of Equation 5.31. These equations will be included in the chapter on efficient color transformations.

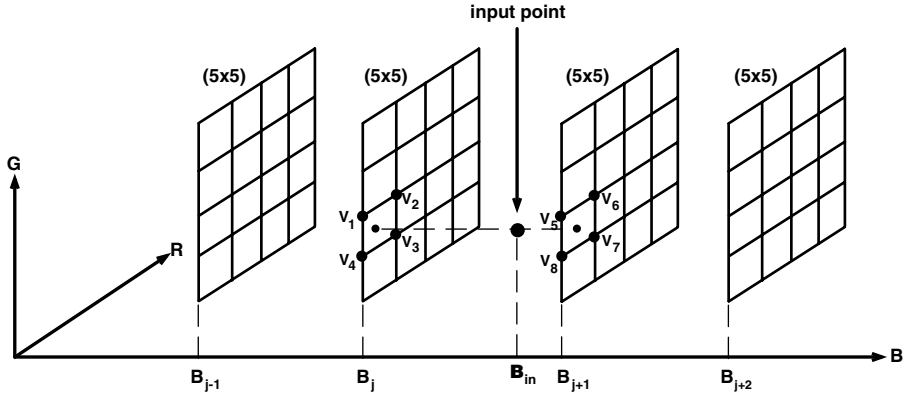
In the context of deriving a characterization function, regular lattices of training data can occur only for the case of output devices, as the patches in the color target can be designed to lie on a regular lattice in device space. With input device characterization, neither the captured device values nor the measured device-independent values of the color target can be guaranteed to lie on a regular lattice.

5.4.6 Sequential interpolation

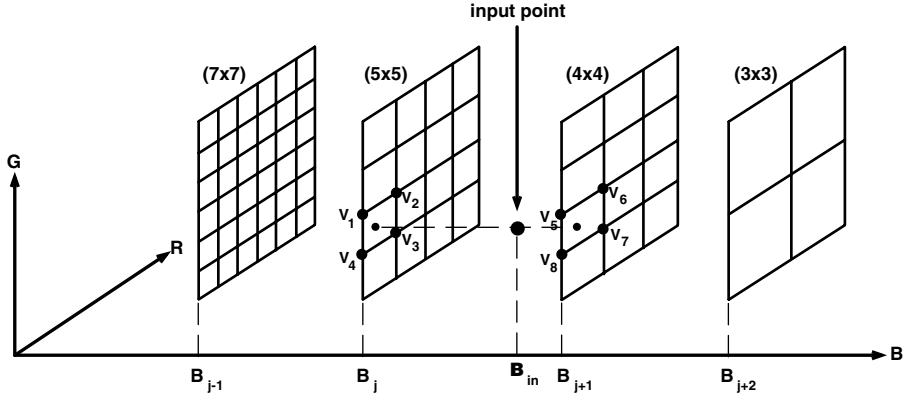
A primary advantage of a regular lattice is that it facilitates simple interpolation techniques. However, it limits the freedom in the placement of control points in multidimensional color space. Referring to Figure 5.12, one would expect considerable curvature of the characterization function in certain regions within the device gamut, while large regions outside the gamut would never be used for interpolation calculations. It would be desirable, therefore, to finely sample regions within the gamut, and coarsely sample regions far away from the gamut. As shown in the figure, the regular lattice does not permit this. A simple extension of regular lattice interpolation, which we term *sequential interpolation (SI)*, brings additional flexibility at a modest increase in computational cost.

In general terms, SI can be thought of as a two-stage interpolation process. Consider a decomposition of the space R^m into two subspaces of dimensions p and q, i.e., $R^m = R^p \times R^q$, $m = p + q$. The m-dimensional lattice L^m can also be decomposed into two sub-lattices L^p and L^q. Let s be the size of L^q. We can think of L^m as being a family of s p-dimensional lattices. In a conventional regular lattice each p-dimensional lattice is identical, and we have $L^m = L^p \times L^q$. In sequential interpolation, we let the p-dimensional lattice structure vary as a function of the remaining q dimensions.

To crystallize this concept, consider the three-dimensional lattice in Figure 5.18 used to implement a characterization function from device RGB to CIELAB. This lattice can be conceived as a family of two-dimensional RG lattices, corresponding to different levels of the third-dimension B. In Figure 5.18a, the RG lattices are identical as a function of B, which corresponds to a regular lattice in RGB space. In this case, interpolation of an input RGB point is accomplished by selecting a subset of the eight vertices $V_1, \ldots, V_8$ that enclose the point and performing a weighted average of the output values at



(a)



(b)

Figure 5.18 Comparison of (a) conventional and (b) sequential interpolation lattices.

these vertices. In Figure 5.18b, a sequential structure is shown where the RG lattice structure is allowed to change as a function of B. The interpolation calculation is accomplished by first projecting an input RGB point onto the B dimension and selecting the neighboring levels B_j and B_{j+1}. These correspond to two lattices in RG space. The input RGB point is then projected onto RG space, and two-dimensional interpolation is performed within each of these lattices, yielding two output colors $\mathbf{c}_j$, $\mathbf{c}_{j+1}$. Finally, one-dimensional interpolation is performed in the B dimension to produce the final output color. In this example, SI would be advantageous if the characterization function is known to exhibit different degrees of curvature for different values of B. If, for example, the function curvature is high for small values of B, SI permits a finer lattice sampling in these regions (as shown in Figure 5.18). Thus, with more efficient node placement, SI enables a given level of accuracy to be achieved with fewer lattice nodes than can be achieved with a regular lattice.

Figure 5.19 is a flow diagram showing the general case of SI in m-dimensions. Application of SI to CMYK printer characterization will be described in Section 5.10.3. Another special case of SI is sequential linear interpolation (SLI).[14] In SLI, we decompose the m-dimensional space into $(m - 1)$ dimensional and one-dimensional subspaces, then decompose the former into $(m - 2)$ and one-dimensional subspaces, and so on until we have a sequence of one-dimensional interpolations. SLI is described in more detail in Chapter 11.

5.4.7 *Neural networks*

Neural networks have taken inspiration from natural computational processes such as the brains and nervous systems of humans and animals. This class of techniques has received much attention in color imaging in recent years. In this section, we briefly describe the use of neural nets in device characterization, referring the reader to Masters[15] for excellent overviews, algorithms, and further reading on the subject.

A neural network is an interconnected assembly of simple processing units called *neurons* whose functionality is loosely based on the biological neuron. The processing ability of the network is stored in the inter-neuron connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns. In the most common configuration, the neurons are arranged into two or more layers, with inputs to neurons in a given layer depending exclusively on the outputs of neurons in previous layers. An example of such a multilayer feed-forward neural network is shown in Figure 5.20. This network has three inputs, three outputs, and one hidden layer of four neurons. The inputs are obtained from an external source (e.g., in our application, color data from the characterization process), and the outputs are the neural network's approximation of the response to these inputs. Let $s_i^{(L)}$ be the ith neuron in the Lth layer, $i = 1, \ldots, N_L$. The output from unit $s_i^{(L)}$ is given by

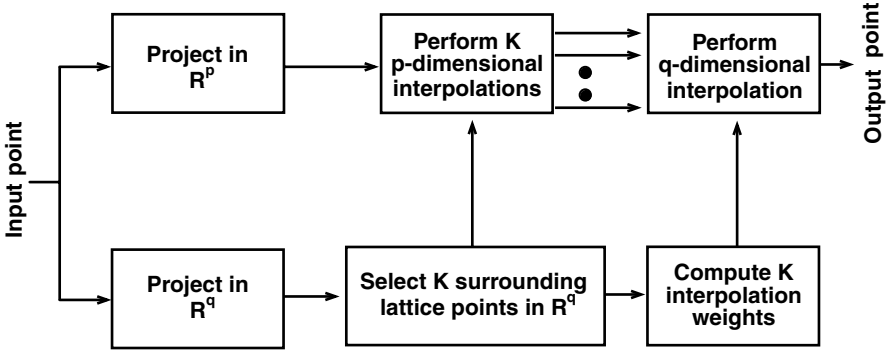


Figure 5.19 Block diagram of sequential interpolation.

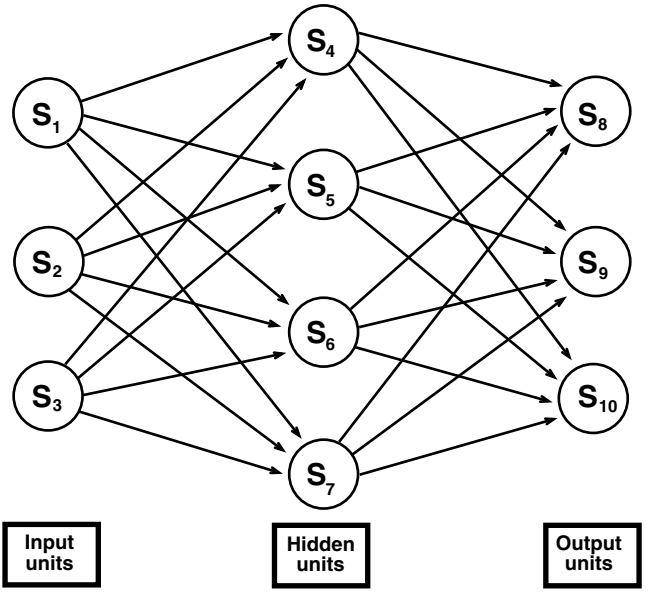


Figure 5.20 Three-layer (3–4–3) neural network.

$$s_i^{(L)} = h\left(\sum_{j=1}^{N_{L-1}} w_{ij} s_j^{(L-1)}\right) \tag{5.33}$$

where w_{ij} = a synaptic weight that determines the relative strength of the contribution of neuron $s_j^{(L-1)}$ to neuron $s_i^{(L)}$

function $h()$ = a nonlinear function, such as a step function or sigmoidal (S-shaped) function

Examples of sigmoidal functions are the logistic function, cumulative Gaussian, and hyperbolic tangent.[15] Depending on the particular architecture being implemented, constraints such as monotonicity and differentiability are often imposed on $h()$. The functionality of the overall neural net is determined by the number of layers and number of neurons per layer, the interconnecting links, the choice of $h()$, and the weights w_{ij}. Note from Equation 5.33 that each layer feeds only to the immediately following layer; this is the most typical configuration.

A popular method for neural network optimization is back-propagation, where all parameters except the synaptic weights w_{ij} are chosen beforehand, preferably based on some *a priori* knowledge about the nature of the function being approximated. The w_{ij} are then derived during a learning process in which a set of training samples in both input and output spaces is presented

to the network. An error metric such as the mean squared error in Equation 5.12 is minimized at the training samples with respect to w_{ij}. Because the overall neural network is a complex nonlinear function of w_{ij}, iterative error minimization approaches are called for. An example is the gradient descent algorithm, where a weight $w_{ij}^{(k)}$ at iteration k is given by

$$w_{ij}^{(k)} = w_{ij}^{(k-1)} - R\left(\frac{\delta E}{\delta w_{ij}}\right) \tag{5.34}$$

Here, E is the error metric being minimized, and R is a parameter known as the learning rate. The iteration continues until some convergence criterion is met with respect to the magnitude or the rate of change of E. The parameter R dictates the speed and stability of convergence. A major shortcoming of the gradient descent algorithm is that convergence is often unacceptably slow. An alternative search technique favored for significantly faster convergence is the conjugate gradient algorithm. As with all iterative algorithms, rate of convergence also depends on the choice of initial estimates, i.e., $w_{ij}^{(0)}$. Linear regression can be used to generate good initial estimates. Details are given in the book by Masters.[15]

The application to color characterization should be evident. A neural network can be used to approximate either the forward or inverse characterization functions. The training samples are the device-dependent and device-independent colors $\{\mathbf{c}_i, \mathbf{d}_i\}$ obtained in the characterization process. After the neural net is trained, arbitrary color inputs can now be processed through the network. The architecture of the network is chosen based on the expected complexity of the characterization function. As with polynomials, increased complexity can result in a better fit up to a certain point, beyond which the network will begin to track the noise in the data.

Typically, the iterative training can be a highly computationally intensive process. Fortunately, this is not a major concern, as this step is carried out offline. Neural networks are also usually too computationally intensive for real-time processing of image pixels. They can, however, be approximated by multidimensional LUTs, which are more computationally efficient.

5.4.8 Spline fitting

Spline interpolation constitutes a rich and flexible framework for approximating free-form shapes. One-dimensional splines can be used very effectively for the calibration step, whereas the multidimensional versions are applicable for characterization. The most common spline functions comprise a set of piecewise polynomial functions defined over a partition of segments in the input space, as shown for the one-dimensional case in Figure 5.21. The behavior of the spline is dictated by control points, known as *knots*, at the segment boundaries. The parameters of the polynomials are determined so that the function passes through all the knots while maintaining certain degrees of continuity across the segment boundaries.

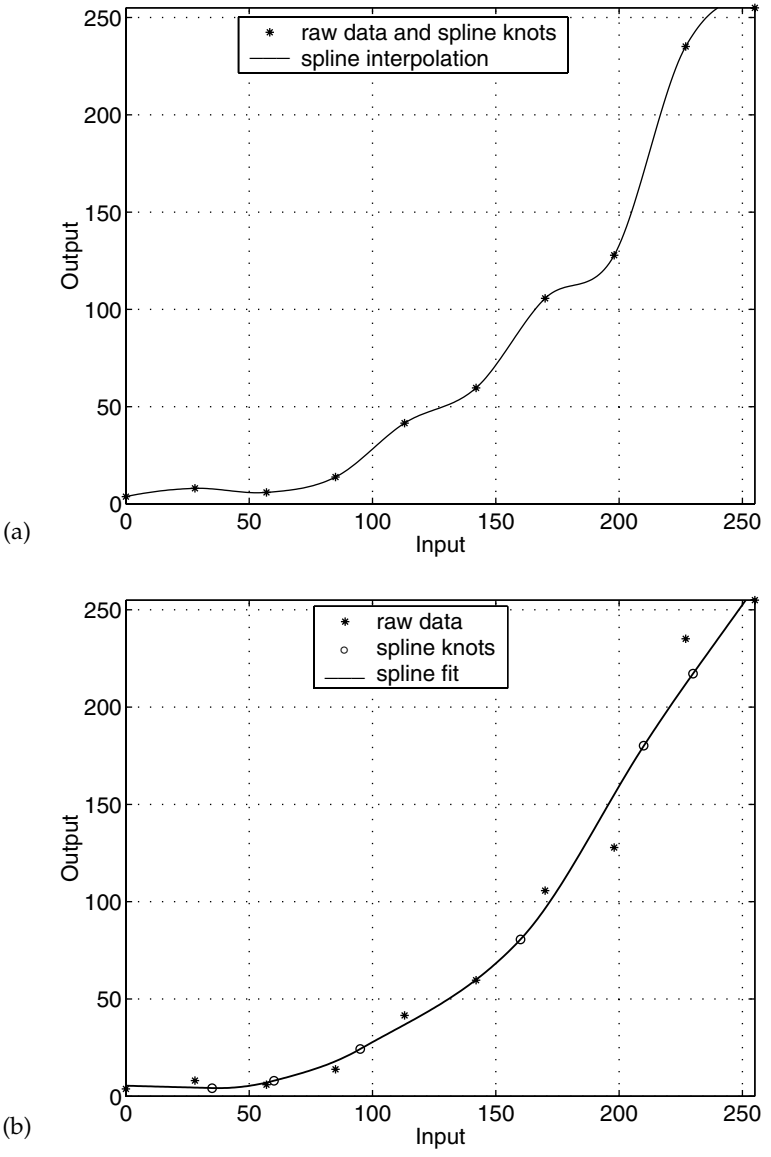


Figure 5.21 Spline function used for (a) interpolation and (b) fitting.

Splines can be used for both interpolation and fitting. In the case of interpolation, shown in Figure 5.21a, the knots coincide with the data points. This approach is desirable when very few accurate data points are available. In the case of fitting, shown in Figure 5.21b, the control points do not necessarily coincide with the data and are actually free parameters chosen to minimize an error criterion between the data points and the spline fit. This approach is preferred when ample data is available but expected to be noisy

and therefore requiring some smoothing. The number and location of the knots used for spline fitting are critical. Too few knots could result in an excessively "stiff" spline that is unable to follow the curvature of the function, but too many knots could result in overshoots that follow the noise. A general guideline is to use fewer knots than data points and to space them approximately uniformly except in regions known to exhibit high curvature, where a denser sampling of knots can be used. As advised earlier, it is highly instructive to first plot and visualize the raw data so as to choose the knots appropriately.

The major advantage of splines over straightforward polynomial approximation is that the complexity of a spline can be tailored to suit the local characteristics of the function. Equivalently, a local change in a calibration or characterization function can be accurately approximated with a change in one local segment of a spline curve. Piecewise cubic and B-splines are popular choices for data fitting applications. Figure 5.22 is a comparison of cubic spline interpolation with the third-order polynomial approximation using the same data as in Figure 5.13. Clearly, the spline is capable of following the data more closely.

Space constraints do not permit a detailed treatment of splines in this chapter. The reader is referred to the book by Farin[16] for a comprehensive tutorial on the subject. C programs for cubic spline interpolation can be found in *Numerical Recipes in C*.[17] Users of Matlab can find an extensive set of spline functions in the spline toolbox (go to www.mathworks.com for details). As with other data-fitting techniques, the most suitable choice of spline function requires knowledge of the nature of the characterization data.

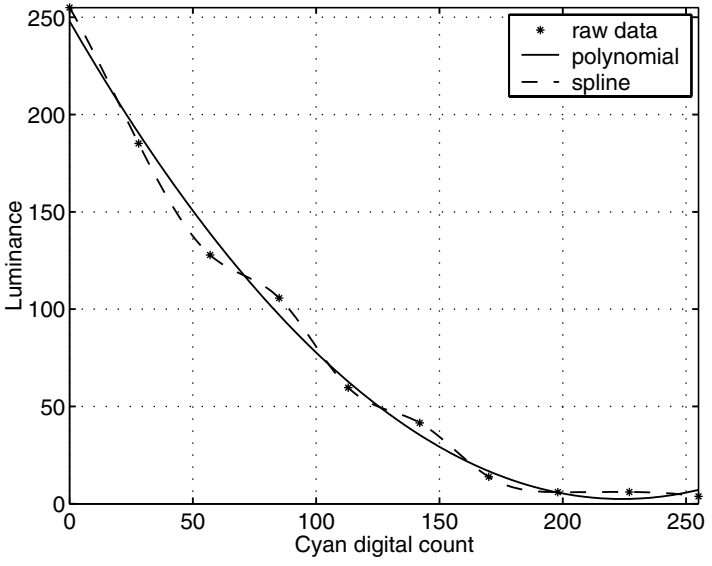


Figure 5.22 Comparison of spline and polynomial fitting.

5.5 Metrics for evaluating device characterization

Many of the mathematical techniques described in the previous section minimize quantitative error metrics. The resulting error from the fitting or interpolation is one indicator of the overall accuracy of characterization. However, this information is not sufficient, for several reasons:

1. The error is available only for the training samples.
2. The error is not always calculated in a visually meaningful color space.
3. Noise and other imperfections that can occur with multiple uses of the device are implicitly ignored.

To address the first concern, the notion of evaluating the characterization with independent test targets was introduced in Section 5.2. To address the second issue, evaluation of errors with visually relevant metrics is strongly recommended. While color difference formulae are described in detail in an earlier chapter, two of them, ΔE^*_{ab} and ΔE^*_{94} are restated here, as they are used extensively in this chapter. Given two CIELAB colors, and their component-wise differences, ΔL^*, Δa^*, Δb^* (equivalently, ΔL^*, ΔC^*, ΔH^*), the ΔE^*_{ab} color difference formula is simply the Euclidean distance between the two points in CIELAB space,

$$\Delta E^*_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} = \sqrt{(\Delta L^*)^2 + (\Delta C^*)^2 + (\Delta H^*)^2} \qquad (5.35)$$

It is important to bear in mind that ΔH^* is not a component-wise hue difference but rather is given by

$$\Delta H^* = \sqrt{(\Delta E^*_{ab})^2 - (\Delta L^*)^2 - (\Delta C^*)^2} \qquad (5.36)$$

The ΔE^*_{94} formula is an extension of ΔE^*_{ab} that applies different weights to the various components as follows:

$$\Delta E^*_{94} = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right)^2} \qquad (5.37)$$

where $S_L = 1$
$S_C = 1 + 0.045\ C^*$
$S_H = 1 + 0.015\ C^*$

The parameters k_L, k_C, and k_H account for the effect of viewing conditions. Under a set of nominal viewing conditions, these parameters are set to 1,

and the overall effect is dictated solely by S_C and S_H, which reduce the perceived color difference as chroma increases.

Another metric used widely in the textile industry is the CMC color difference formula. This formula is similar in form to the ΔE^*_{94} equation and has parameters tailored for perceptibility vs. acceptability of color differences. Finally, an extension of the ΔE^*_{94} formula has been recently developed, known as the CIEDE2000 metric.[18] This metric accounts for interactions between the C^* and H^* terms and is expected to be adopted as an industry standard until further developments arise. The reader is referred to Chapter 1 for details.

The next question to consider is what error statistics to report. Common aggregate statistics cited in the literature are the mean, standard deviation, minimum, and maximum of the ΔEs for a set of test samples. Often, a cumulative statistic such as the 95th percentile of ΔE values (i.e., the value below which 95% of the ΔE values in the test data lie) is calculated. For a complete statistical description, histograms of ΔE can also be reported.

Having chosen an error metric, how does one determine that the characterization error is satisfactorily small? First, recall that characterization accuracy is limited by the inherent stability and uniformity of a given device. If the errors are close to this lower bound, we know that we cannot do much better for the given device. In the following sections, we will provide the reader with some idea of the characterization accuracy achievable by state-of-the-art techniques. It must be kept in mind, however, that "satisfactory accuracy" depends strongly on the application and the needs and expectations of a user. A graphic arts color proofing application will likely place stringent demands on color accuracy, while inexpensive consumer products will typically play in a market with wider color tolerances.

Another aspect that further confounds evaluation of color accuracy is that the end user ultimately views not test targets with color patches but images with complex color and spatial characteristics. Unfortunately, quantitative analysis of patches is not always a reliable indicator of perceived color quality in complex images. (The latter is a subject of active research.[19]) The reader is thus advised to exercise appropriate caution when interpreting individual results or those cited in the literature, and to always augment quantitative evaluation of color accuracy with a qualitative evaluation involving images and individuals that represent the intended market and application.

A special class of error metrics for input devices evaluates how accurately the information recorded by the input device can be transformed into the signals sensed by the human visual system for input stimuli with given spectral statistics. Such error metrics do not directly evaluate the accuracy of a characterization but rather the ability of the device to act as a visual colorimeter. Hence, these metrics are relevant for filter design optimization and can also suggest the most appropriate characterization technique for a given input device. The reader is referred to papers by Sharma et al.[20] and Quan et al.[21] for further details.

5.6 Scanners

All scanners employ one of two primary types of sensing technology. Drum scanners use photomultiplier tubes (PMTs), whereas the less expensive flatbed scanners employ charge-coupled devices (CCDs). Both of these technologies sense and convert light input into analog voltage. Drum scanners consist of a removable transparent cylinder on which a print, which is reflective, transparent, or a photographic negative, can be mounted. A light source illuminates the image in a single pass as the drum spins at a high speed. The light reflected off or transmitted through the print is passed through red, green, and blue filters then sent through the PMTs, which relay voltages proportional to the input light intensity. The PMT is extremely sensitive, thus providing drum scanners a large dynamic range. The drum scanners used in offset printing applications contain built-in computers that are capable of direct conversion of the RGB scan to CMYK output and are used to generate color separations at very high spatial resolution. A limitation of this scanning technology is that the original must be flexible so that it can physically be mounted on the drum.

All flatbed scanners utilize CCD technology, which is simpler, more stable, and less costly than PMT technology. These scanners have widely varying sensitivity and resolution and, at the highest end, approach the performance of drum scanners. Transparent or reflective prints are placed on a glass platen and evenly illuminated from above the glass for transparencies, and from beneath for reflective. As the light source moves across the image, individual lines of the image are sensed by a CCD array, which relays voltages that are proportional to the input light intensity. An integrating cavity is usually employed to focus light from the scanner illuminant onto the print. An undesirable outcome of this is that light reflected from a given spatial location on the print can be captured by the cavity and returned to the print at neighboring locations. Hence, the scanner measurement at a pixel depends not only on the reflectance at that pixel but also on the reflectances of neighboring pixels. A model and correction algorithm for this so-called *integrating cavity effect* is given by Knox.[22]

Following the sensing step, an analog-to-digital (A/D) converter is used to quantize the analog voltage signal to a digital signal represented by between 8 and 16 bits per each of R, G, B channels. These raw digital values are usually linear with respect to the luminance of the stimulus being scanned. Additional image acquisition software often allows the raw data to be processed through tone reproduction curves so that a power-law (or gamma) relationship exists between digital value and luminance. This operation is carried out before the A/D conversion. One reason for doing this is that quantization of nonlinear gamma-corrected signals is less visually disturbing than quantization of data that is linear in luminance. (This is discussed in more detail in the section on display characterization.) A second reason is to prepare the scanned data for direct display on a CRT, which exhibits approximately a square law (gamma = 2) relationship.

5.6.1 *Calibration*

Scanner calibration involves first establishing various settings internal to the scanner, or in the scanner driver. To calibrate the white point, a reflective white sample shipped with the scanner is scanned, and the gain factor on each of the R, G, B signals is adjusted so that R = G = B = 1 for this sample. As mentioned earlier, additional scanner software can offer selections for the digital precision of the RGB output and transformations between analog and digital representations (e.g., power-law functions). Once set, these parameters must not be altered during subsequent characterization or scanning operations.

In addition, it is usually desirable to linearize and gray-balance the scanner response. The result of this step is that an input ramp of gray stimuli in equal increments in luminance will result in equal increments in R = G = B scanner values. To achieve this, the scanner is exposed to a ramp of gray patches of known luminance values (e.g., as found at the bottom of the Q60 target); the scanner RGB values are extracted for each patch, and a TRC is constructed. A hypothetical example is given in Figure 5.23 to illustrate the process. The TRC is constructed so that a triplet of raw RGB values corresponding to a gray patch will map to the corresponding measured luminance value (within a scaling factor). The measurements generally provide only a subset of the data points in the TRC, the rest being determined with some

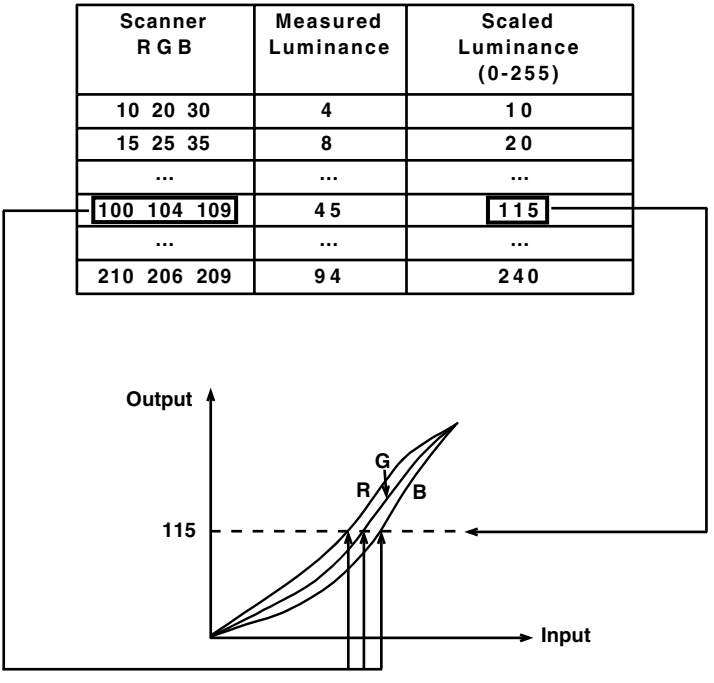


Figure 5.23 Illustration of gray-balance calibration for scanners.

form of data fitting or interpolation technique. Because the data are likely to contain some noise from the scanning and measuring process, it is preferable that the fitting technique incorporate some form of smoothing. Kang[23] reports that linear regression provides sufficiently accurate results for scanner gray balance, while nonlinear curve fitting offers only a modest improvement. In any event, polynomial and spline techniques are viable alternatives for scanners that exhibit significant nonlinearity.

5.6.2 Model-based characterization

Model-based scanner characterization attempts to establish the relationship between calibrated device-dependent data and colorimetric representations via explicit modeling of the device spectral sensitivities. Adopting the notation in previous sections, consider a training set of T spectral reflectance samples $\{\mathbf{s}_i\}$, which can be collected into a matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_T]^t$. The spectral data is related to device data $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_T]^t$ and colorimetric data $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_T]^t$ by Equations 5.1 and 5.2, respectively. In matrix notation, we thus have

$$\mathbf{C} = \mathbf{S}\mathbf{A}_c \;;\; \mathbf{D} = \mathbf{S}\mathbf{A}_d \tag{5.38}$$

The column vectors of matrix $\mathbf{A}_c$ are a product of the color matching functions and the viewing illuminant $\mathbf{I}_v$, and similarly $\mathbf{A}_d$ is formed from a product of the scanner spectral sensitivities and the scanner illuminant $\mathbf{I}_s$. The classic model-based approach is to compute the linear 3×3 matrix transformation $\mathbf{M}$ that best fits the colorimetric data to device-dependent data in the least-squared error sense. The linear approximation is expressed as

$$\mathbf{C} \approx \mathbf{D} \cdot \mathbf{M} \tag{5.39}$$

and from Section 5.4.1, the optimal $\mathbf{M}$ is the least-squares solution,

$$\mathbf{M} = (\mathbf{D}^t\mathbf{D})^{-1}\,\mathbf{D}^t\mathbf{C} \tag{5.40}$$

Plugging Equation 5.38 into Equation 5.40, we have

$$\mathbf{M} = (\mathbf{A}_d^t\mathbf{S}^t\,\mathbf{S}\mathbf{A}_d)^{-1}\,\mathbf{A}_d^t\mathbf{S}^t\,\mathbf{S}\mathbf{A}_c \tag{5.41}$$

Equation 5.41 tells us that the scanner characterization function is determined by

1. Color matching functions
2. Viewing and scanning illuminants $\mathbf{I}_v$ and $\mathbf{I}_s$
3. Spectral autocorrelation matrix $\mathbf{S}^t\mathbf{S}$ of the training samples
4. Scanner spectral sensitivities

Note that Equation 5.40 can be directly used to estimate M from a set of training samples $\{\mathbf{d}_i, \mathbf{c}_i\}$ without explicit knowledge of the spectral sensitivities. However, for accurate results, this empirical procedure would have to be repeated for each different combination of input reflectances $\mathbf{S}$ and viewing illuminants $\mathbf{I}_v$. The model-based formulation, Equation 5.41, allows prediction of the scanner response for arbitrary input reflectances and illuminants given the scanner sensitivities $\mathbf{A}_d$ and illuminant $\mathbf{I}_s$. The optimal $\mathbf{M}$ can be computed using Equation 5.41 without having to make repeated measurements for every combination of input media and illuminants.

Each of the quantities of interest in Equation 5.41 will now be discussed. Because the color matching functions $\mathbf{A}_c$ are known functions, they are not included in the discussion.

Viewing illuminant. In general, it is difficult to ascertain *a priori* the illuminant under which a given stimulus will be viewed. A common *de facto* assumption for viewing reflective prints is the Daylight 5000 (D50) illuminant. However, if it is known that images are to be viewed under a certain type of lighting, e.g., cool-white fluorescence or an incandescent lamp, then the corresponding spectral radiance should be used.

Scanning illuminant. Scanners typically employ a fluorescent source, hence the spectral radiance function will contain sharp peaks as shown in Figure 5.4. The spectral radiance function $I_s(\lambda)$ can be obtained from the scanner manufacturer or can be estimated from the training data. However, the peaks found in fluorescent sources can lead to unreliable estimates unless these are explicitly modeled.[24] Hence, it is generally preferable that this quantity be directly measured.

Scanner spectral sensitivities. Deriving the scanner sensitivities is the most challenging aspect of model-based characterization. Some scanner manufacturers supply such data with their products. However, the information may not be accurate, as filter characteristics often change with time and vary from one scanner to another. Direct measurement of the scanner sensitivities may be achieved by recording the scanner response to narrowband reflectance data. However, this is a difficult and expensive process and therefore impractical in most applications. The most viable alternative is to estimate the sensitivities from a training set of samples of known spectral reflectance. Several approaches exist for this and are briefly described below, along with references for further reading.

The most straightforward technique is to use least-squares regression to obtain the device sensitivity matrix $\mathbf{A}_d$. The objective is to find $\mathbf{A}_d$ that minimizes $\|\boldsymbol{D} - \boldsymbol{S}\boldsymbol{A}_d\|^2$. From the linear regression formulation in Section 5.4.2, we have

$$\mathbf{A}_d = (\mathbf{S}^t\,\mathbf{S})^{-1}\,\mathbf{S}^t\mathbf{D} \tag{5.42}$$

The problem with this approach is that, although the spectral reflectance data is L-dimensional, with L being typically between 31 and 36, the true dimensionality of the spectra of samples found in nature is significantly less.

(Studies have shown that the samples in the Macbeth chart can be accurately represented with as few as three basis functions.[25]) Alternatively phrased, the system of Equations 5.42 contains only a small number of significant eigenvalues. This results in the spectral autocorrelation matrix $\mathbf{S}^t\mathbf{S}$ being ill conditioned, in turn yielding unstable, noise-sensitive estimates of the sensitivity functions $\mathbf{A}_d$. One approach to mitigate this problem is to use only the eigenvectors corresponding to the few most significant eigenvalues of $\mathbf{S}^t\mathbf{S}$ in the solution of Equation 5.42. This so-called "principal eigenvector" (PE) method results in a solution that is far less noise sensitive than that obtained from Equation 5.42. The reader is referred to Sharma[24] for more details.

One problem with PE is that it does not exploit *a priori* information about the nature of the spectral sensitivity functions. We know, for example, that the spectral sensitivities are positive-valued and usually single-lobed functions. In the case where $\mathbf{A}_d$ only contains the passive filter and detector responses (i.e., the illuminant is not included), we also know that the functions are smooth. There are a number of ways to use these constraints to generate estimates of $\mathbf{A}_d$ that are superior to those achieved by PE. One approach is to define the aforementioned constraints as a set of linear inequalities and formulate the least-squares minimization as a quadratic programming problem. The latter can be solved using standard packages such as Matlab. The reader is referred to Finlayson et al.[26] for more details. Another approach is to use a set theoretical formulation to express the constraints as convex sets and to use an iterative technique known as *projection onto convex sets (POCS)* to generate the sensitivity functions.[24] One potential problem with the POCS technique is that the solution is not unique and is often sensitive to the initial estimate used to seed the iterative process. Despite this caveat, this technique has been shown to produce very good results.[24,27]

Input spectral data. As alluded to in Section 5.2, the spectral reflectance data $\mathbf{S}$ should be measured from media that are representative of the stimuli to be scanned. If a single scanner characterization is to be derived for all possible input media, it is advisable to measure the data from a wide range of media, e.g., photography, offset, laser, inkjet, etc. An interesting case occurs if $\mathbf{S}$ is constructed by drawing samples at random from the interval [–1, 1] with equal likelihood. With this "maximum ignorance" assumption, the spectral data are uncorrelated; therefore, the autocorrelation $\mathbf{S}^t\mathbf{S}$ is an identity matrix, and Equation 5.41 reduces to

$$\mathbf{M} = (\mathbf{A}_d{}^t\mathbf{A}_d)^{-1}\,\mathbf{A}_d{}^t\mathbf{A}_c \tag{5.43}$$

Note that the characterization transform now no longer depends on measured data. Observe, too, that Equation 5.43 is also the least-squares solution to the linear transformation that relates the color matching functions $\mathbf{A}_c$ to the device sensitivities $\mathbf{A}_d$;

$$\mathbf{A}_c \approx \mathbf{A}_d\mathbf{M} \tag{5.44}$$

Comparing Equations 5.39 and 5.44, we see that the optimal linear transform that maps the color matching functions to the scanner sensitivities is the same as the transform that optimally maps scanner RGB to XYZ under the maximum ignorance assumption. As a corollary, if the scanner is perfectly colorimetric, then Equations 5.39 and 5.44 become equalities, and the matrix that relates the color matching functions to scanner sensitivities is precisely the matrix that maps scanner RGB to XYZ for all media and illuminants.

One problem with the maximum ignorance assumption is that it includes negative values, which can never occur with physical spectra. Finlayson et al.[28] show that a positivity constraint on the preceding formulation results in the correlation $\mathbf{S}^t\mathbf{S}$ being a constant (but not identity) matrix, which results in a more accurate estimate of $\mathbf{M}$.

Another class of model-based techniques, somewhat distinct from the preceding framework, derives scanner characterization for a specific medium by first characterizing the medium itself and using models for both the medium and scanner to generate the characterization. The additional step of modeling the medium imposes physically based constraints on the possible spectra S and can lend further insight into the interaction between the medium and the scanner. Furthermore, *a priori* modeling of the input medium may simplify the *in situ* color measurement process. Berns and Shyu[29] postulate that scanner filters are designed to align closely with the peaks of the spectral absorptivity functions of typical photographic dyes. The relationship between scanner RGB and C, M, Y dye concentrations is thus modeled by simple polynomial functions. The Beer–Bouguer and Kubelka–Munk theories (discussed in Section 5.10.2) are then used to relate dye concentrations to reflectance spectra for photographic media. Sharma[30] models the color formation process on photographic media using the Beer–Bouguer model. From this model, and using a small number of measurements on the actual sample being scanned, the set $\mathbf{S}_{medium}$ of all reflectance spectra reproducible by the given medium is estimated. For a given scanner RGB triplet, the set $\mathbf{S}_{scanner}$ of all reflectance spectra that can generate this triplet is derived with knowledge of the scanner spectral sensitivities, $\mathbf{A}_d$. The actual input reflectance spectrum lies in the intersection $\mathbf{S}_{medium} \cap \mathbf{S}_{scanner}$ and is derived using POCS. Note that both these approaches generate spectral characterizations, i.e., mappings from scanner RGB to spectral reflectance. From this, colorimetric characterizations can readily be generated for arbitrary viewing illuminants.

5.6.3 *Empirical characterization*

Empirical approaches derive the characterization function by correlating measured CIE data from a target such as the Q60 to scanned RGB data from the target. Most of the data-fitting techniques described in Section 5.4 can be used (with the exception of lattice-based approaches, as scanner

characterization data cannot be designed to lie on a regular grid). Kang[23] describes the use of polynomial regression to fit gray-balanced RGB data to CIEXYZ measurements. He compares 3×3, 3×6, 3×9, 3×11, and 3×14 polynomial matrices derived using least-squares regression as described in Section 5.4.3. Several targets, including the MacBeth ColorChecker and Kodak Q60, are used. The paper concludes that a 3×6 polynomial offers acceptable accuracy and that increasing the order of the polynomial may improve the fit to training data but may worsen the performance on independent test data. This is because, as noted in Section 5.4, higher-order approximations begin to track the noise in the data. The paper also explores media dependence and concludes that the optimal 3×3 matrix does not vary considerably across media, whereas the optimal polynomial transform is indeed media dependent and will generally offer greater accuracy for any given medium.

Kang and Anderson[31] describe the use of neural networks for scanner characterization. They use a 3–4–3 network, trained by cascaded feed-forward correlation. A cumulative Gaussian function is used for the nonlinearity at each unit in the network (see Section 5.4.7). In comparison with polynomial regression, the neural network reports superior fits to training data but inferior performance for independent test data. Furthermore, the neural network is reported as being fairly sensitive to the choice of training data. Hence, while neural networks offer powerful capabilities for data fitting, much care must be exercised in their design and optimization to suit the nature of the particular device characteristics.

5.7 Digital still cameras

Digital still cameras (DSCs) are becoming a common source for digital imagery. Their characterization is complicated by two factors.

1. The conditions under which images are captured are often uncontrolled and can vary widely.
2. To compensate for this, DSC manufacturers build automatic image-processing algorithms into the devices to control and correct for flare, exposure, color balance, etc.

DSC characterization is probably unnecessary in most consumer applications and is called for only in specialized cases that require controlled, high-quality color capture. In such cases, it is imperative that the automatic processing be disabled or known to the extent that the raw DSC signals can be recovered from the processed data.

A few precautions are in order for proper digital capture of calibration and characterization targets. First, it must be ensured that the illumination on the target is uniform. A viewing/illuminating geometry of 0/45 is recommended so as to be consistent with the geometry of measurement devices and typical visual viewing of hardcopy prints. Next, the lenses in most digital

cameras do not transmit light uniformly across the lens area, so, for a fixed input radiance, pixels near the center report higher signal levels than those in the periphery. The ideal solution to this problem is to expose the camera to a constant color (e.g., gray) target and digitally compensate for any spatial uniformity in the camera response. (Such compensation may be built into some camera models.) The effect can also be somewhat reduced by choosing the distance between camera and target so that the target does not occupy the full camera frame. Finally, it is recommended that any nonvisible radiation to which the DSC is sensitive be blocked so that output RGB values are not affected. Many DSCs respond to IR radiation, hence IR blocking filters should be used.

Figure 5.24 shows the color calibration and characterization path for a DSC. Much of the theoretical framework for image capture is common between DSCs and scanners; hence, we will frequently refer to the formulation developed in Section 5.6 for scanners while focusing here on DSC-specific issues. For additional procedural details on DSC characterization, the reader is referred to the ISO 17321 standard.[32]

5.7.1 Calibration

It must be ensured that camera settings such as aperture size and exposure time are in a known fixed state, and that all automatic color processing is disabled. The main task in DSC calibration is to determine the relationship between input scene radiance and camera response, typically for a range of gray input stimuli. Determination of this function, known as the opto-electronic conversion function (OECF), is conceptually similar to the gray-balancing operation for a scanner (see Section 5.6.1). A target comprising gray patches of known spectral reflectance measurements is illuminated with a known reference illuminant. From the reflectance and illuminant data, the luminance Y of each patch is calculated (see Equation 5.9). An image of the target is captured with the DSC. The correspondence between input luminance Y and output RGB is used to generate an inverse OECF function as described in Section 5.6.1 for scanners. This is a TRC that maps raw device

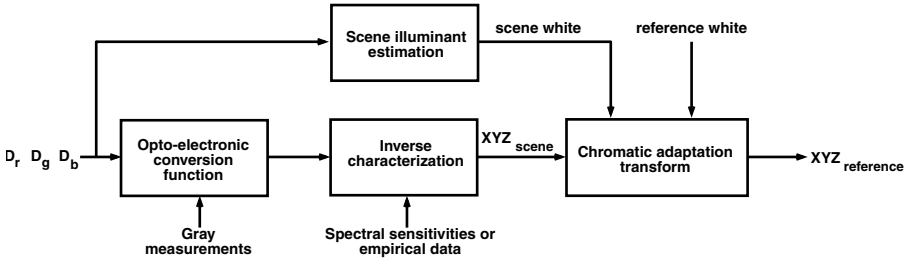


Figure 5.24 Block diagram of digital camera calibration, characterization, and chromatic adaptation transforms.

RGB to R′G′B′ such that R′ = G′ = B′ = Y for the neutral patches. The raw captured image is then always processed through this TRC to obtain a linearized and gray-balanced image prior to subsequent processing. Further details on specifications and experimental procedures for OECF determination are given in the ISO 14524 standard.[33]

5.7.2 Model-based characterization

The goal is to obtain the optimal 3×3 matrix **M** that relates the DSC RGB data to a colorimetric (e.g., XYZ) representation. As with scanners, derivation of **M** is given by Equation 5.41 and requires knowledge of the color matching functions, correlation statistics of scene data, and device spectral sensitivities. Color matching functions are known and require no further discussion. Scene correlation statistics should be used where possible. However, given the diversity of scene content likely to be encountered by a DSC, the maximum ignorance assumption is often invoked, and scene statistics are eliminated from the formulation. Derivation of **M** thus reduces to Equation 5.43 and requires only estimation of the DSC spectral sensitivities.

The techniques described in Section 5.6.2 for estimating device sensitivities indirectly from the characterization data can be applied for DSCs. One can also adopt a more direct approach of recording the device's response to incident monochromatic light at different wavelengths. The latter can be generated by illuminating a diffuse reflecting surface with light filtered through a monochromator. From Equation 5.1, the camera response to monochromatic light at wavelength λ is given by

$$D_i(\lambda) = I_m(\lambda)R_d(\lambda)q_i(\lambda) = S(\lambda)q_i(\lambda) \tag{5.45}$$

where $i = R, G, B$
$I_m(\lambda)$ = the monochromator illumination
$R_d(\lambda)$ = the reflectance of the diffuse surface
$S(\lambda) = I_m(\lambda)R_d(\lambda)$ is the radiance incident to the DSC

For simplicity, the detector sensitivity $u(\lambda)$ in Equation 5.1 is folded into the term $q_i(\lambda)$ in Equation 5.45, and the noise term is assumed to be negligible. The radiance $S(\lambda)$ is measured independently with a spectroradiometer. The spectral sensitivities $q_i(\lambda)$ are then obtained by dividing the camera response $D_i(\lambda)$ by the input radiance $S(\lambda)$. In the case where the DSC response is tied to a specific reference illuminant $I_{ref}(\lambda)$, the products $q_i(\lambda)I_{ref}(\lambda)$ can be stored. More details are found in ISO 17321.[32]

The reader is reminded that, due to practical considerations, DSC sensitivities are not linearly related to color matching functions, and that the 3×3 matrix being derived is only an approximation. However, this approximation is sufficient for many applications. The accuracy of **M** for critical colors can be further improved by imposing constraints on preservation of white and neutral colors.[34]

5.7.3 Empirical characterization

As with scanners, empirical DSC characterization is accomplished by directly relating measured colorimetric data from a target and corresponding DSC RGB data obtained from a photographed image of the target. This approach is recommended in the case where the DSC spectral sensitivities are unknown, or when the target and illumination conditions used for characterization are expected to closely match those encountered during actual image capture.

Hubel et al.[35] compare several techniques for computing the optimal 3×3 matrix **M**. One of these is a model-based approach that uses a white point preserving maximum ignorance assumption, while the remaining techniques are empirical, using linear regression on training samples. They report an extensive set of results for different illumination conditions. Average ΔE_{CMC} values range from approximately 2.5 to 6.5, depending on the technique and illumination used. The model-based technique was often outperformed by an empirical technique for a given medium and illuminant. However, the model-based strategy, being oblivious to scene statistics, was generally robust across different illumination conditions.

An empirically derived characterization need not be restricted to a linear transformation. Hong et al.[36] explore a polynomial technique to characterize a low-performance Canon PowerShot Pro70 camera for photographic input. A second-order polynomial was employed with 11 terms given by [D_r, D_g, D_b, D_rD_g, D_rD_b, D_gD_b, D_r^2, D_g^2, D_b^2, $D_rD_gD_b$, 1]. The average characterization error for 264 training samples from an IT8.7/2 target was $\Delta E_{CMC(1:1)} = 2.2$. A similar technique[37] was used to characterize a high-performance Agfa digital StudioCam resulting in an average $\Delta E_{CMC(1:1)} = 1.07$. Note that these errors are significantly lower than those reported by Hubel et al. This is not surprising, because polynomials can be expected to outperform linear approximations under a given set of controlled characterization conditions. The other findings from these two studies are as follows:

- Correction for the OECF significantly improves overall characterization accuracy.
- For polynomial fitting, 40 to 60 training samples seem adequate; beyond this, there is little to be gained in characterization accuracy.
- The polynomial correction is highly dependent on the medium/colorant combination.
- For a single medium/colorant combination, increasing the order of the polynomial up to 11 improves the characterization accuracy, with some terms (notably $D_rD_gD_b$ and the constant term) being more important than others. With the high-performance camera, a 3×11 polynomial results in an average error of approximately 1 $\Delta E_{CMC(1:1)}$. The low-performance camera results in $\Delta E_{CMC(1:1)} = 2.2$.
- For cross-media reproduction, increasing the order of the polynomials is not of significant benefit. Typical accuracy with a 3×11

correction lies between 2 and 4 $\Delta E_{CMC(1:1)}$ when characterization and test media are not the same.

5.7.4 White-point estimation and chromatic adaptation transform

The characterization step described in Sections 5.7.2 and 5.7.3 yields a transformation between DSC data and colorimetric values corresponding to the input viewing conditions. One must be able to convert this colorimetric data to a standard color space (e.g., sRGB), which is based on a different set of reference viewing conditions. This calls for a color appearance model to account for the differences between input and reference viewing conditions. The most important parameters pertaining to the viewing conditions are the input scene and reference white points. The appearance model can thus be reduced to a chromatic adaptation transform (CAT) between the two white points.

In general, the scene white is unknown and must be indirectly estimated from the image data. A recent technique, known as color by correlation, has shown promise as a simple and reliable method of estimating white point. The idea is to acquire *a priori* sets of DSC training data corresponding to different known illuminants. Data from a given image are then compared with each training set, and the illuminant is chosen that maximizes the correlation between the image and training data. If the DSC spectral sensitivities are known, the training samples can be acquired via simulation; otherwise, they must be gathered by photographing samples under different illuminants. See Chapter 5 of Reference 7 for details of this approach.

There has been considerable research in finding the optimal color space for the CAT. An excellent survey is given in Chapter 5 of Reference 7. Ideally, the CAT should mimic visual adaptation mechanisms, suggesting that it should be performed in an LMS cone fundamental space. Finlayson et al.[38] use the added argument that orthogonal visual channels maximize efficiency to orthogonalize the LMS space, forming their so-called *sharp* color space. (The term "sharp" comes from the fact that the associated color matching functions are relatively narrowband.) Psychophysical validation has shown that the sharp space is among the best spaces for performing the CAT. A physically realizable variant of this space is being proposed as an ISO standard for DSC characterization.[32] This ISO-RGB space is a linear transformation of XYZ, and is given by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4339 & 0.3762 & 0.1899 \\ 0.2126 & 0.7152 & 0.0721 \\ 0.0177 & 0.1095 & 0.8728 \end{bmatrix} \begin{bmatrix} R \\ G \\ G \end{bmatrix}; \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} \begin{bmatrix} 3.0799 & -1.5369 & -0.5432 \\ -0.9209 & 1.8756 & 0.0454 \\ 0.0531 & -0.2041 & 1.1510 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}; \tag{5.46}$$

The procedure for applying the CAT in ISO-RGB space given the input and reference white points is summarized as follows:

1. Use the calibration and characterization transforms to convert DSC device data to XYZ_{in} corresponding to input viewing conditions.
2. Convert the input and reference white points from XYZ to ISO-RGB using Equation 5.46.
3. Convert XYZ_{in} to ISO-RGB_{in} using Equation 5.46.
4. Perform von Kries chromatic adaptation by multiplying ISO-RGB_{in} by the diagonal matrix,

$$\begin{bmatrix} \frac{R_{ref}^{white}}{R_{in}^{white}} & 0 & 0 \\ 0 & \frac{G_{ref}^{white}}{G_{in}^{white}} & 0 \\ 0 & 0 & \frac{B_{ref}^{white}}{B_{in}^{white}} \end{bmatrix} \tag{5.47}$$

   where C_{ref}^{white}, C_{in}^{white}, $(C = R, G, B)$ are the white points under reference and input viewing conditions, respectively. (The reader is referred to an earlier chapter for details on von Kries adaptation.) This step generates ISO-RGB data under reference viewing conditions, denoted ISO-RGB_{ref}.
5. Convert ISO-RGB_{ref} to XYZ_{ref} using Equation 5.46. This provides a colorimetric representation under reference viewing conditions and can be transformed to other standard color spaces.

Note that the matrices in the last three steps can be concatenated into a single 3×3 matrix for efficient processing.

5.8 CRT displays

The cathode-ray tube (CRT) is the most common type of display used in computers and television. Color is produced on a CRT display by applying modulated voltages to three electron guns, which in turn strike red, green, and blue phosphors with electrons. The excited phosphors emit an additive mixture of red, green, and blue lights. The assumptions mentioned in Section 5.2.4 on channel independence and chromaticity constancy, in addition to the usual assumptions on spatial uniformity and temporal stability, result in a fairly simple process for CRT calibration and characterization.

5.8.1 Calibration

Cathode-ray tube (CRT) calibration involves setting brightness and contrast controls on the display to a fixed nominal value. In addition, the relationship between the R, G, B input digital values driving the three gun voltages and

the resulting displayed luminance must be established and corrected. This relationship is usually modeled based on the power-law relationship between the driving voltage and the beam-current for a vacuum tube, and is given by[39]

$$\frac{Y_R}{Y_{R_{max}}} = \begin{cases} f + K_R\left(\dfrac{D_R - D_{offset}}{D_{max} - D_{offset}}\right)^{\gamma_R} & \text{if } D_R > D_{offset} \\ f & \text{if } D_R \le D_{offset} \end{cases} \tag{5.48}$$

where D_R = input digital value to the red gun
Y_R = resulting luminance from the red channel
$Y_{R_{max}}$ = luminance of the red channel at full intensity
D_{offset} = largest digital count for which there is no detectable luminance from the screen
D_{max} = maximum digital count (e.g., in an 8-bit system, $D_{max} = 255$)
f = flare that arises mostly from ambient illumination
K_R = a gain factor
γ_R = nonlinear power law factor

Analogous expressions hold for the green and blue terms. Generally, the calibration is done with all room lights turned off; hence, the flare term is assumed to be negligible. In addition, with proper brightness and contrast settings, the following simplifying assumptions are often made: $K_R = K_G = K_B = 1$, $D_{offset} = 0$. This reduces Equation 5.48 to

$$\frac{Y_R}{Y_{R_{\max}}} = \left(\frac{D_R}{D_{\max}}\right)^{\gamma_R} \tag{5.49}$$

with analogous expressions for Y_G and Y_B. The parameters for the calibration model are obtained by making measurements of a series of stepwedges from each primary color to black using a spectroradiometer or colorimeter and fitting these measurements to the model given by Equations 5.48 or 5.49 using regression. If Equation 5.49 is adopted, a simple approach is to take logarithms of both sides of this equation to produce a linear relationship between $\log(Y_R)$ and $\log(D_R/D_{max})$. This can then be solved for γ_R via the linear regression technique described in Section 5.4.1. Berns et al.[39] provide detailed descriptions of other regression techniques. Values of γ_R, γ_G, γ_B for typical CRTs lie between 1.8 and 2.4.

Once the model is derived, a correction function that inverts the model is applied to each of the digital R, G, B inputs. If Equation 5.49 is assumed, the correction is given by

$$D_R = D_{\max}\left(\frac{D'_R}{D_{\max}}\right)^{1/\gamma_R} \tag{5.50}$$

with similar expressions for G and B. Here D'_R, D'_G, D'_B are linear in luminance, and D_R, D_G, D_B are the raw signals that drive the gun voltages. The calibration function, Equation 5.50, is often referred to as gamma correction and is usually implemented as a set of three one-dimensional lookup tables that are loaded directly into the video path. Plots of Equations 5.49 and 5.50 for $\gamma = 1.8$ are shown in Figure 5.25.

It is worth noting that digital quantization of the gamma-corrected signal D_R, D_G, D_B in Equation 5.50 results in smaller quantization intervals at lower luminance values where the eye is more sensitive to errors, and larger intervals at high luminances where the eye is less sensitive. The idea of applying nonlinear preprocessing functions to reduce the visual perceptibility of quantization errors (often known as *companding*) is widely employed in many digital signal processing applications. In our case, gamma correction applied prior to conversion to the digital domain not only calibrates the CRT, it also fortuitously reduces perceived quantization error in color images intended for CRT display.

The CRT with the gamma correction Equation 5.50 incorporated in the video path exhibits a tone reproduction characteristic that is linear in luminance. That is,

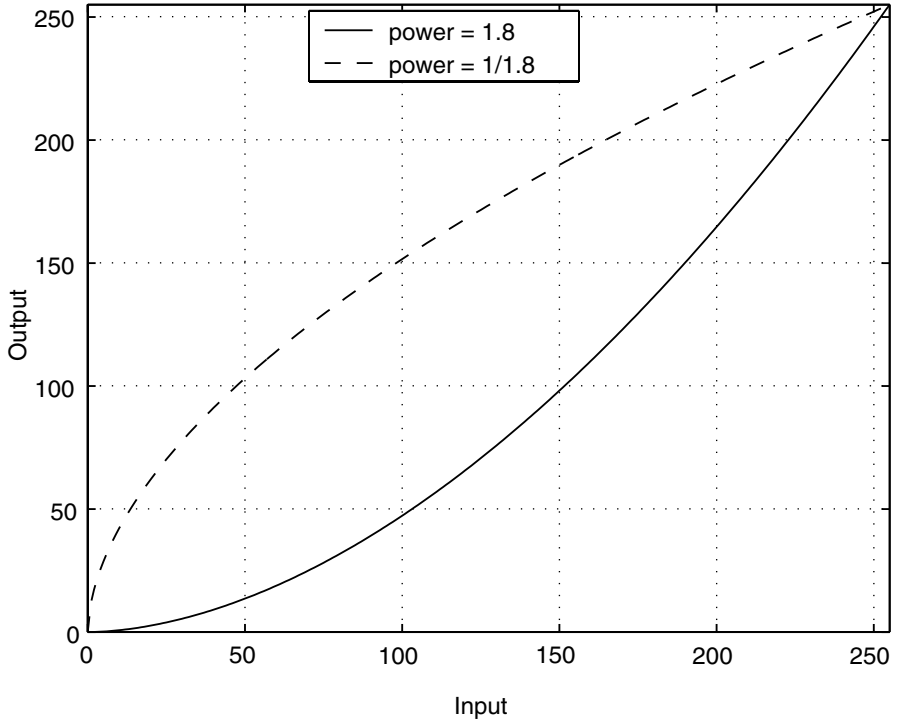


Figure 5.25 Gamma function for $\gamma = 1.8$.

$$\frac{Y_R}{Y_{R_{\text{Max}}}} = \left(\frac{D'_R}{D_{\text{max}}}\right) \tag{5.51}$$

with similar expressions for G and B. Some CRT calibration packages allow the user to specify an overall system gamma, γ_{system}, so that Equation 5.51 becomes

$$\frac{Y_R}{Y_{R_{max}}} = \left(\frac{D'_R}{D_{max}}\right)^{\gamma_{system}} \tag{5.52}$$

This provides some control on the effective tone reproduction characteristic of the CRT. To achieve this overall system response, the gamma correction function Equation 5.50 is modified as

$$D_R = D_{max}\left(\frac{D'_R}{D_{max}}\right)^{\left(\frac{\gamma_{system}}{\gamma_R}\right)} \tag{5.53}$$

5.8.2 Characterization

We assume henceforth that the aforementioned calibration has been derived so that Equation 5.51 holds. Recall that, with the assumptions on channel independence and chromaticity constancy, Equation 5.8 describes the relationship between input device RGB values and output spectral radiance. Spectral radiance is then converted to tristimulus XYZ values according to Equation 5.9. Substituting the expression for $S_{RGB}(\lambda)$ in Equation 5.8 into Equation 5.9., the relationship between the inputs D'_R, D'_G, D'_B to a linearized CRT and resulting tristimulus values is given by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{bmatrix} \begin{bmatrix} D'_R \\ D'_G \\ D'_B \end{bmatrix} \tag{5.54}$$

where X_R, Y_R, Z_R = tristimulus values of the red channel at its maximum intensity, and likewise for green and blue

In matrix-vector notation, Equation 5.54 becomes

$$\mathbf{c} = \mathbf{A}_{CRT}\,\mathbf{d'}; \qquad \mathbf{d'} = \mathbf{A}_{CRT}^{-1}\,\mathbf{c} \tag{5.55}$$

The columns of $\mathbf{A}_{CRT}$ are the tristimulus coordinates of R, G, B at maximum intensity and can be obtained by direct tristimulus measurement. A more robust approach would be to include additional tristimulus measurements of other color mixtures and to solve for $\mathbf{A}_{CRT}$ using least-squares regression as described in Section 5.4.1. Note that $\mathbf{A}_{CRT}$ assumes flare-free viewing conditions. If flare is present, this can be captured in the **d'** vector by using calibration function Equation 5.48 with an appropriate value for *f*.

The quality of the characterization can be evaluated by converting a test set of color patches specified in XYZ to display RGB through the inverse characterization mapping (i.e., the second part of Equation 5.55) and measuring the displayed colors (see Figure 5.10). The original and measured values are then converted to CIELAB coordinates, and the error is derived using a suitable metric such as ΔE^*_{ab} or ΔE^*_{94}. Berns et al.[39] report excellent results using this simple model, with average ΔE^*_{ab} less than 1. Factors that can contribute to additional errors include internal flare within the CRT, cross-channel interactions not accounted for in the aforementioned model, and spatial nonuniformity across the display.

Most CRTs exhibit fairly similar color characteristics, because the power-law relationship, Equation 5.48, is a fundamental characteristic of vacuum tube technology; furthermore, CRT manufacturers use very similar, if not identical, phosphor sets. For this reason, and because CRTs are such a prevalent medium for the display and manipulation of color, there have been several efforts to standardize on CRT RGB color spaces. The most notable recent example is the sRGB standard (available at www.srgb.com). If one's CRT has not been characterized, one of the standard models can be adopted as a reasonable approximation. Minimally, these RGB spaces are defined by a gamma (assumed to be equal for all channels) and matrix $\mathbf{A}_{CRT}$. Sometimes, instead of directly specifying $\mathbf{A}_{CRT}$, the *x*-*y* chromaticity coordinates of the red, green, and blue primaries are provided along with the XYZ values of the white point. $\mathbf{A}_{CRT}$ is easily derived from these quantities (see Appendix 5.B).

5.8.3 *Visual techniques*

Because CRTs can be accurately characterized with simple models, a class of techniques has emerged that obviates the need for color measurements and relies upon visual judgments to directly estimate model parameters such as gamma and offset.[40–42] The basic idea is to display a series of targets on the screen and provide the user with some control to adjust certain colors until they match given reference stimuli. Based on the settings selected by the user, an algorithm computes the model parameters. An example is shown in Figure 5.26 for visually determining γ in Equation 5.49. The bottom half of the target is a fine checkerboard pattern of alternating black and white dots. The top half is a series of patches at different gray levels. The user is asked to select the gray patch whose luminance matches the average luminance of the checkerboard. The assumption is that the average checkerboard luminance $Y_{checkerboard}$ is approximately halfway between the luminances of

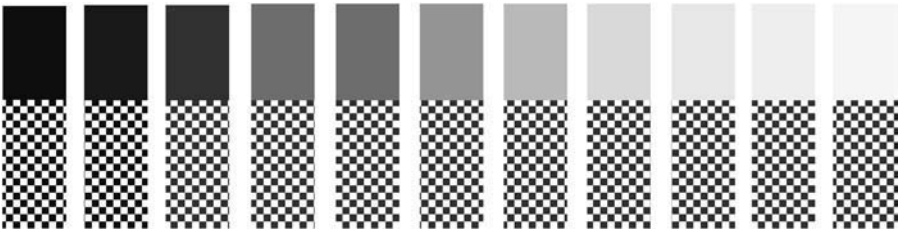


Figure 5.26 Target for visual determination of γ for displays.

black and white. Reasonable *a priori* assumptions can be made for the latter (e.g., $Y_{black} = 0$ and $Y_{white} = 100$, respectively), and hence for the checkerboard (e.g., $Y_{checkerboard} = 50$). A user who selects the gray patch to match the checkerboard is effectively selecting the digital count D_{match} corresponding to luminance $Y_{checkerboard}$. This provides enough information to calculate γ by rearranging Equation 5.49 as follows:

$$\gamma = \frac{\log(Y_{checkerboard}/Y_{white})}{\log(D_{match}/D_{max})} \tag{5.56}$$

In this example, the same γ value is assumed for the R, G, and B channels. The technique is easily extended to estimate γ for each individual channel by displaying checkerboard patterns that alternate between black and each respective primary. A demonstration of visual CRT calibration can be found in the recent article by Balasubramanian et al.[43] Visual determination of the color of the primaries and white point (i.e., $\mathbf{A}_{CRT}$) requires more sophisticated techniques[44] and is an active area of research.

5.9 Liquid crystal displays

Liquid crystal displays are becoming an increasingly popular medium for color display. Their compactness and low power consumption, combined with steadily increasing spatial resolution and dynamic range, have made these devices increasingly prevalent in both consumer and professional markets. Consequently, color management for LCDs has received greater attention in recent years.

The type of LCD most commonly used for computer display is the backlit active-matrix LCD (AMLCD) employing twisted nematic technology. In this technology, each pixel comprises a pair of linear polarizers and a liquid crystal substrate sandwiched between them. The polarizations are oriented orthogonally to each other. Light from a source behind the display surface passes through the first polarizer and is then reoriented by the liquid crystal substrate before it is passed through the second polarizer. The light then passes through one of red, green, or blue filters, arranged in a spatial mosaic. The extent of optical reorientation by the liquid crystal, and thus the intensity

of light finally emanated, is determined by an electric field applied to the liquid crystal substrate. This field is determined by an applied voltage, which in turn is driven by the digital input to the device.

From the viewpoint of color characterization, twisted nematic technology can pose several shortcomings: the strong dependence of perceived color on viewing angle, poor gray balance for R = G = B input, and lack of chromaticity constancy. Recent developments such as in-plane switching technology[45] overcome these problems to some extent.

5.9.1 *Calibration*

A major difference between CRT and LCD characteristics is the nonlinear function that relates input digital values to output luminance, shown in Figure 5.27. Unlike vacuum tubes that exhibit a power-law relationship, LCD technology results in a native electro-optic response that is often better modeled as a sigmoidal S-shaped function.[46] However, many LCD manufacturers build correction tables into the video card that result in the LCD response mimicking that of a CRT (i.e., a power-law response with $\gamma = 1.8$ or 2.2). Hence, it is recommended that some initial analysis be performed before a particular function is chosen and that, if possible, built-in corrections be deactivated so as to reliably calibrate the raw display response. As with CRTs, the calibration function is derived by making color measurements of a series of stepwedges in each of R, G, B. If a model-based approach is adopted, the model parameters are fitted to the measurements via regression.

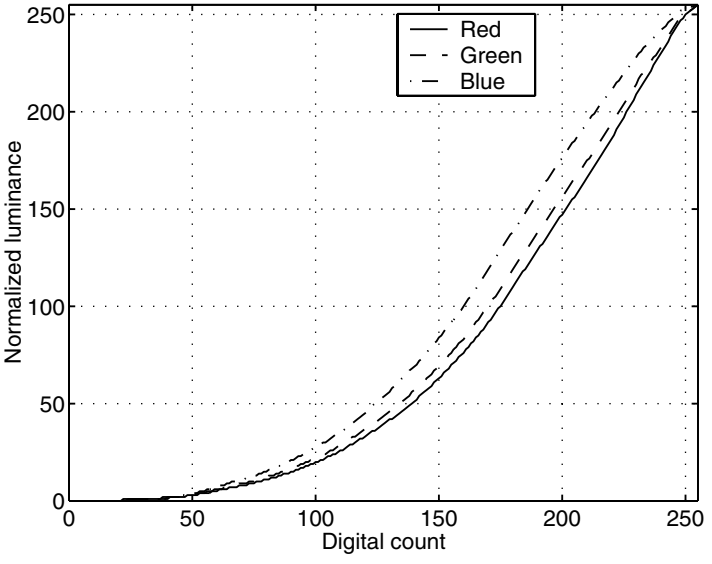


Figure 5.27 Typical opto-electronic conversion function for liquid crystal displays.

Alternatively, if the LCD response does not appear to subscribe to a simple parametric model, an empirical approach may be adopted wherein the measured data are directly interpolated or fitted using, for example, piecewise linear, polynomial, or spline functions.

As mentioned earlier, some LCDs do not adhere to the chromaticity constancy assumption. This is largely due to the non-smooth spectral characteristics of the backlight and its interaction with the color filters.[45] Kwak et al.[47] compensate for the lack of chromaticity constancy by introducing cross terms in the nonlinear calibration functions to capture interactions among R, G, and B. They claim a significant improvement in overall accuracy as a result of this extension.

5.9.2 *Characterization*

Most of the assumptions made with CRTs (i.e., uniformity, stability, pixel independence, and channel independence) hold to a reasonable degree with AMLCDs as well. Hence, the characterization function can be modeled with a 3×3 matrix as in Equation 5.54, and the procedure described in Section 5.8 for deriving CRT characterization can be used for AMLCDs in the same manner. As mentioned earlier, an important caution for AMLCDs is that the radiance of the emanated light can be a strong function of the viewing angle. The only practical recommendation to mitigate this problem is that the measurements should be taken of light emanating perpendicular to the plane of the screen. The same geometry should be used for viewing images. For further details on LCD characterization, the reader is referred to the works by Marcu,[45] Sharma,[46] and Kwak.[47]

5.10 *Printers*

Printer characterization continues to be a challenging problem due to the complex nonlinear color characteristics of these devices. Space considerations do not permit a description of the physics of the numerous digital printing technologies. Instead, we will offer general techniques that apply to broad categories of devices (e.g., halftone vs. continuous tone; CMY vs. CMYK, etc.).

Recall the basic calibration and characterization workflow in Figure 5.9. The techniques for target generation and calibration and characterization vary widely, offering a range of trade-offs between cost and accuracy. A selection of common techniques will be presented in this section.

5.10.1 *Calibration*

Two common approaches are channel-independent and gray-balanced calibration.

5.10.1.1 Channel-independent calibration

In this type of calibration, each channel i (i = cyan, magenta, yellow, etc.) is independently linearized to a defined metric M_i. An example of such a metric is the ΔE^*_{ab} color difference between the ith channel and medium white, defined as

$$M_i(d) = \|\mathbf{c}_{medium} - \mathbf{c}_i(d)\|_2, \quad i = C,M,Y, 0 \le d \le d_{max} \tag{5.57}$$

where d = input digital level
$\mathbf{c}_{medium}$ = CIELAB measurement of the medium
$\mathbf{c}_i(d)$ = CIELAB measurement of the ith colorant generated at digital level d

Note that, by definition, $M_i(0) \equiv 0$. Linearizing with respect to this metric will result in an approximately visually linear printer response along each of its primary channels.

The calibration is accomplished with the following steps:

- Generate stepwedges of pure C, M, Y patches at a few selected digital levels d_j. The number and spacing of levels required depend on the characteristics of the printer. As a general guideline, between 15 and 20 patches per channel is sufficient for most printers, and a finer sampling is recommended in the region of small d values to accurately capture the printer response at the highlights. Also ensure that the solid patch (i.e., $d = d_{max}$) is included.
- Make CIELAB measurements of the stepwedges and of the bare medium. Media relative colorimetry is recommended for the CIELAB calculations.
- Evaluate $M_i(d_j)$ at the measured digital levels d_j using Equation 5.57.
- Scale the data by a multiplicative factor so that $M_i(d_{max}) = d_{max}$. This is accomplished by multiplying the function $M_i(d)$ by the constant $[d_{max}/M_i(d_{max})]$.
- Invert the scaled functions $M_i(d)$ to obtain M_i^{-1} by interchanging the dependent and independent variables. Use some form of fitting or interpolation to evaluate M_i^{-1} for the entire domain $[0, d_{max}]$. If the printer response is smooth, linear interpolation suffices; otherwise, more sophisticated fitting techniques such as polynomials or splines are called for (see Section 5.4.). The result is the calibration function, which can be implemented as a set of one-dimensional TRCs for efficient processing of images.
- Test the calibration by running a stepwedge of uniformly spaced digital values of a single colorant through the TRC, printing and measuring the resulting patches, and computing M_i. A linear relationship should be achieved between the digital input to the TRC and the resulting M_i. Repeat this step for each colorant.

An example of the response $M_i(d)$ for a Xerox DocuColor 12 xerographic printer is shown in Figure 5.28a for 16 digital levels. The scaled $M_i(d)$ are shown in Figure 5.28b. The inverse function is shown in Figure 5.29 and is the final calibration TRC for the DC12. Note that the calibration is essentially a reflection of the printer response $M_i(d)$ about the 45° line. To test the

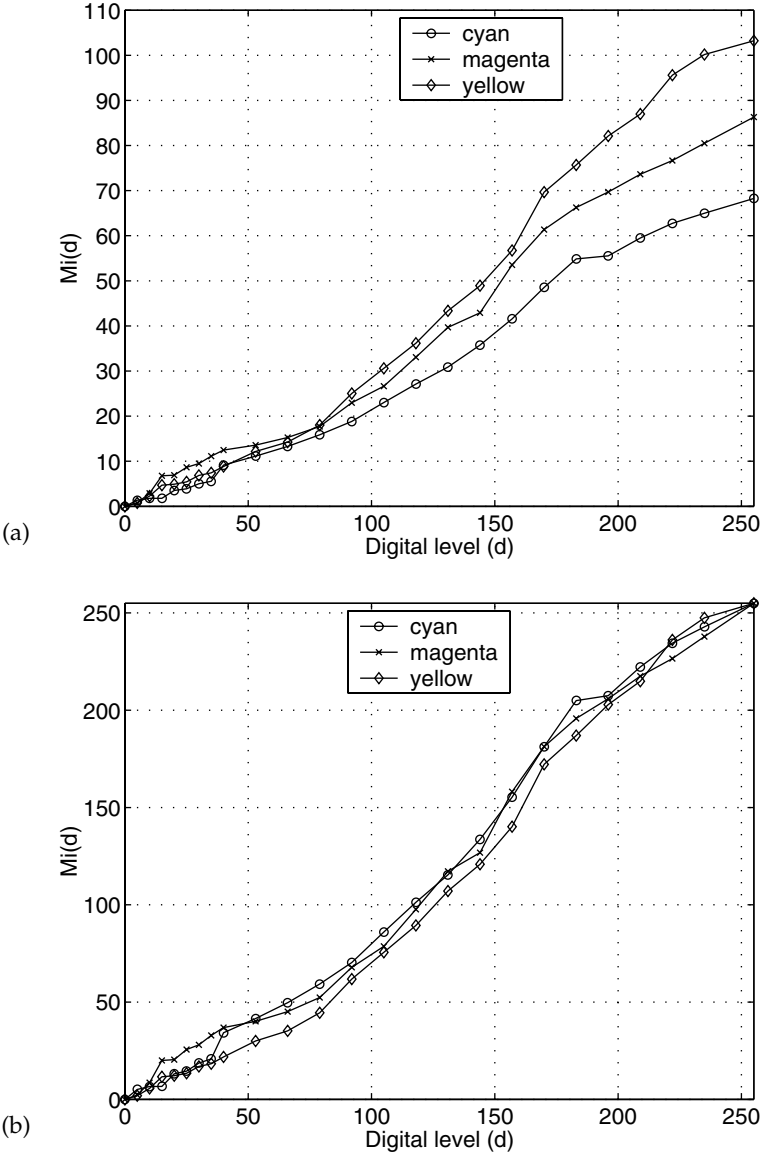


Figure 5.28 Raw device response, $M_i(d)$ defined as ΔE^*_{ab} from paper, for Xerox DocuColor 12 printer: (a) unscaled and (b) scaled to d_{max}.

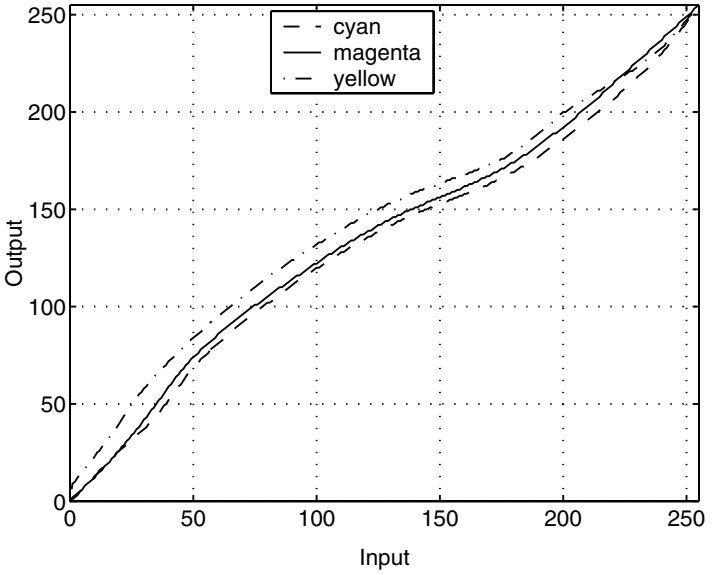


Figure 5.29 Calibration curves correcting for response of Fig 5.28.

calibration, the same C, M, Y stepwedge data were processed through the calibration TRCs, printed, and measured, and M_i was evaluated using Equation 5.57 and plotted in Figure 5.30. The calibrated response is now linear with respect to the desired metric M_i.

Other metrics can be used instead of Equation 5.57, e.g., optical density, or luminance.[48] The calibration procedure is identical.

5.10.1.2 Gray-balanced calibration

An alternative approach to calibration is to gray balance the printer so that equal amounts of C, M, Y processed through the calibration result in a neutral (i.e., $a^* = b^* = 0$) response. There are two main motivations for this approach. First, the human visual system is particularly sensitive to color differences near neutrals; hence, it makes sense to carefully control the state of the printer in this region. Second, gray balancing considers, to a first order, interactions between C, M, and Y that are not taken into account in channel-independent calibration. However, gray balancing is more complicated than channel-independent linearization and generally demands a larger number of patch measurements.

In addition to determining the relative proportions of C, M, Y that generate neutral colors, gray balancing can also achieve a specified tone response along the neutral axis (e.g., linear in neutral luminance or lightness). The following procedure can be used to gray balance and linearize the printer to neutral lightness L^*:

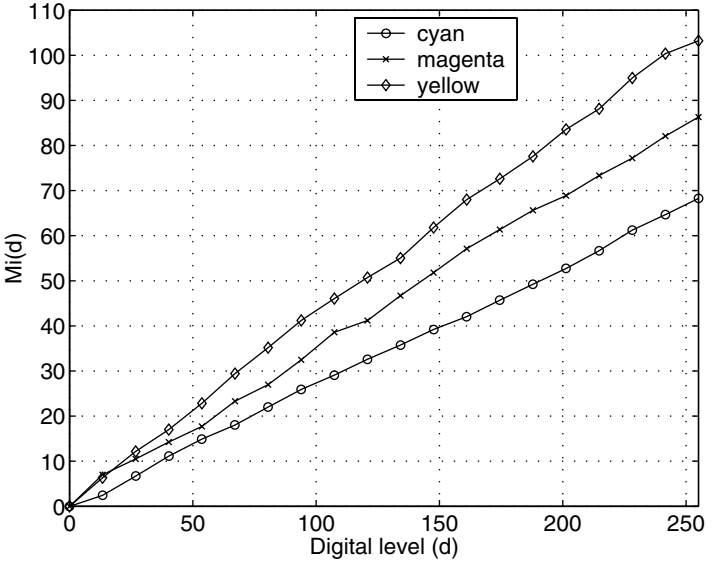


Figure 5.30 Response $M_i(d)$ of calibrated device.

1. Generate a training set of device-dependent (CMY) data in the vicinity of neutrals across the dynamic range of the printer. One exemplary approach is to vary C and M for fixed Y, repeating for different levels of Y across the printer range. The number and spacing of steps for C, M, and Y should be chosen so as to bracket the neutral $a^* = b^* = 0$ axis. Therefore, these parameters depend on printer characteristics, and their selection will require some trial and error.
2. Generate device-independent (CIELAB) data corresponding to these patches. This can be accomplished either via direct measurement of a target containing these patches or by processing the CMY data through a printer model that predicts the colorimetric response of the printer. (Printer models are discussed in a subsequent section.) Media-relative colorimetry is recommended for the CIELAB calculations. If the CIELAB measurements do not bracket the neutral axis, it may be necessary to iterate between this and the previous step, refining the choice of CMY points at a given iteration based on the CIELAB measurements from the previous iteration.
3. Given the training CMY and CIELAB data, obtain CMY values that yield neutral measurements, i.e., $a^* = b^* = 0$, at a set of lightness levels L_i^*, $i = 1, \ldots, T$, spanning the range of the printer. A sufficiently fine sampling of measurements may allow the neutral points to be directly selected; however, in all likelihood, some form of fitting or interpolation will be required to estimate neutral points. A possible candidate is the distance-weighted linear regression function from

Section 5.4.4.2. The regression is supplied with the training data as well as a set of input neutral CIELAB points (L_i^*, 0, 0). The output from the regression is a set of weighted least-squares estimates (C_i, M_i, Y_i) that would produce (L_i^*, 0, 0). A hypothetical example is shown in Figure 5.31a. Typically, 6 to 10 L^* levels are sufficient to determine gray-balance throughout the printer's dynamic range.

4. To generate a monotonically increasing calibration function from Figure 5.31a, invert the sense of the lightness values L_i^* to obtain neutral "darkness" values, denoted D_i^*, scaled to the maximum digital count d_{max}. The formula is given by

$$D_i^* = \left(\frac{d_{\max}}{100}\right)(100 - L_i^*) \tag{5.58}$$

5. Group the data into three sets of pairs $\{D_i^*, C_i\}$, $\{D_i^*, M_i\}$, $\{D_i^*, Y_i\}$, and from this generate three functions, *C(D*), M(D*), Y(D*)* using a one-dimensional fitting or interpolation algorithm. The use of splines is recommended, as these have the flexibility to fit data from a wide variety of printers and also possess the ability to smooth out noise in the data. These functions are plotted in Figure 5.31b for the same hypothetical example. Note that, above a certain darkness $D_{maxgray}^*$, it is not possible to achieve neutral colors, because one of the colorants (cyan in this example) has reached its maximum digital value. Hence, there are no real calibration data points in the input domain $[D_{maxgray}^*, d_{max}]$. One approach to complete the functions is to pad the calibration data with extra values in this region so that the spline

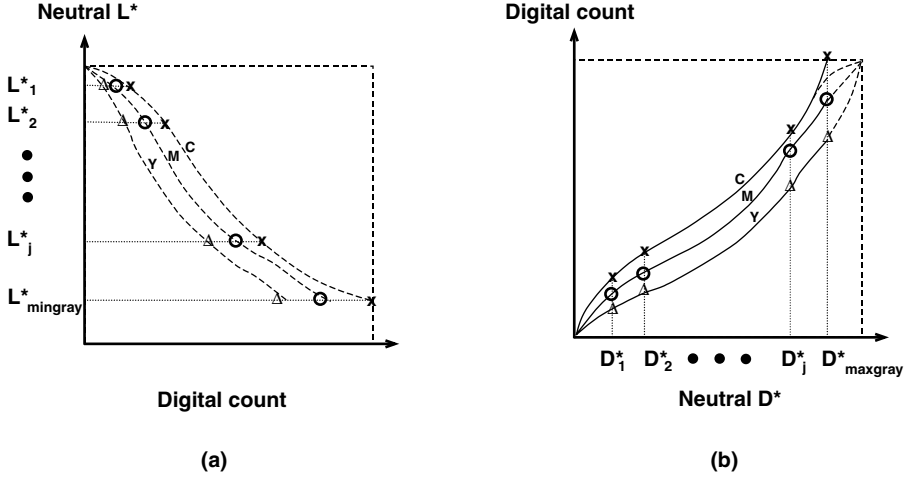


Figure 5.31 Illustration of gray-balance calibration for printers: (a) L^* vs. digital count for neutral samples, and (b) corresponding TRC.

fitting will smoothly extrapolate to the endpoint, d_{max}. Figure 5.31b shows schematically the extrapolation with dashed lines. To achieve smooth calibration functions, it may be necessary to sacrifice gray balance for some darkness values less than $D^*_{maxgray}$. In the case of CMYK printers, the trade-off can be somewhat mitigated by using the K channel in combination with C, M, Y to achieve gray balance. Trade-offs between smoothness and colorimetric accuracy are frequently encountered in printer calibration and characterization. Unfortunately, there is no universal solution to such issues; instead, knowledge of the particular printer and user requirements is used to heuristically guide the trade-offs.

6. Test the calibration by processing a stepwedge of samples C = M = Y = d through the TRCs, and printing and measuring CIELAB values. As before, it is convenient to assess the outcome by plotting L^*, a^*, b^* as a function of the input digital count d. For most of the range $0 \le d \le D^*$, the calibration should yield a linear response with respect to L^*, and $a^*, b^* \approx 0$. If the deviation from this ideal aim is within the inherent variability of the system (e.g., the stability and uniformity of the printer), the calibration is of satisfactory accuracy. Recall that, for gray levels darker than D^*_{max}, a linear gray-balanced response is no longer achievable; instead, the printer response should smoothly approach the color of the CMY solid overprint.
7. An additional test, highly recommended for gray-balance calibration, is to generate a target of continuous ramps of C = M = Y, process through the calibration, print, and visually inspect the output to ensure a smooth neutral response. The prints must be viewed under the same illuminant used for the CIELAB calculations in the calibration.

A recent study by the author has shown that visual tolerance for gray in reflection prints is not symmetric about the $a^* = b^* = 0$ point.[49] In fact, people's memory of and preference for gray occurs in the quadrant corresponding to $a^* < 0$, $b^* < 0$. In regions corresponding to positive a* or b*, a dominant hue is more readily perceived; hence, tolerances for gray reproduction in these regions are small. Colloquially phrased, people prefer "cooler" (bluish/greenish) grays to "warmer" (reddish/yellowish) grays. This observation can exploited to improve the robustness of gray balancing for printers. The device could be balanced toward preferred gray (a*, b* < 0) rather than colorimetric gray (a* = b* = 0) with the same procedure described above. The expected advantage is that, by setting the calibration aim-point in a region with large visual tolerance, the errors inevitably introduced by the calibration are less likely to be visually objectionable.

5.10.2 Model-based printer characterization

Several physics-based models have been developed to predict the colorimetric response of a printer. Some of the common models will be described in

this section. Some overlap exists between this section and an earlier chapter on the physics of color. That chapter focuses on modeling the interaction between light, colorants, and medium at the microscopic level. The emphasis here is in modeling the printer at a macroscopic level, with the goal of deriving the forward characterization mapping from device colorant values to device-independent coordinates such as spectral reflectance or CIEXYZ. Derivation of the inverse characterization function is generally independent of the forward model and is thus discussed in a separate section.

To set a framework for the models to follow, it is instructive to examine different ways in which light passes through a uniform colorant layer. These are depicted in Figure 5.32. In Figure 5.32a, light passes though the colorant layer in only one direction. Some of the light is absorbed, and the remaining is transmitted. The absorption and transmission are functions of wavelength, hence the perception that the layer is colored. This layer is said to be transparent and occurs when the colorant particles are completely dissolved in the medium. The dyes in a dye-diffusion printer can be reasonably well approximated by this model. In Figure 5.32b, some of the light is transmitted and some absorbed as in Figure 5.32a. However, due to the presence of discrete particles, some of the light is also scattered. This layer is said to be translucent. Xerographic and certain inkjet printers subscribe to this model. In Figure 5.32c, a much higher presence of discrete particles results in all of the light being either absorbed or scattered. This layer is said to be opaque, and it applies to paints and some inkjet processes. In all cases, transmission, absorption, and scattering are functions of wavelength. We will see shortly that models for predicting the color of uniform colorant layers are based on one of these three scenarios. More details are given in Chapter 3, which focuses on the physics of color.

In the ensuing discussions, we will assume the exemplary case of a three-colorant (CMY) printer. Extension to an arbitrary number of colorants is usually straightforward.

5.10.2.1 Beer–Bouguer model

The Beer–Bouguer (BB) model plays an important role in colorant formulation, being frequently used to predict light transmission through colored

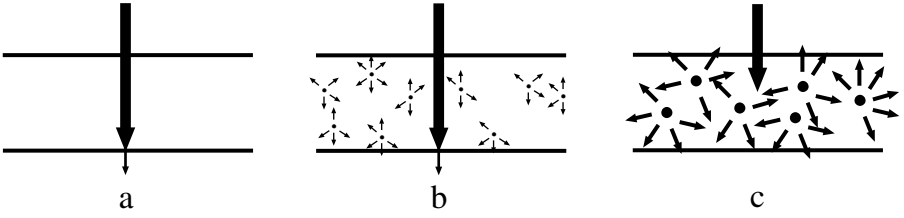


Figure 5.32 Light transport models for (a) transparent, (b) translucent, and (c) opaque media.

materials in liquid solutions. In digital color imaging, it is most applicable for continuous-tone printing with transparent colorants and media (i.e., Figure 5.32a). The underlying assumption is that the spatial rate of change of light radiance as it passes through an absorbing colorant layer is proportional to the radiance itself. Mathematically, this is given by

$$\frac{dI_x(\lambda)}{dx} = -A(\lambda)I_x(\lambda) \tag{5.59}$$

where $I_x(\lambda)$ = radiance at position x within the colorant layer
$A(\lambda)$ = a proportionality factor given by

$$A(\lambda) = \xi\, w\, k(\lambda) \tag{5.60}$$

where ξ = concentration of the colorant
w = thickness of the colorant layer
$k(\lambda)$ = spectral absorption coefficient of the colorant

Substituting Equation 5.60 into Equation 5.59 and integrating with respect to x over the thickness of the colorant, we obtain the following expression for the radiance $I(\lambda)$ emerging from the colorant:

$$I(\lambda) = I_0(\lambda)\exp(-\xi w k(\lambda)) = I_i(\lambda)T_t(\lambda)\exp(-\xi w k(\lambda)) \tag{5.61}$$

where $I_0(\lambda)$ is the radiance of light that would be transmitted in the absence of the colorant, which can be expressed as the product of the incident light $I_i(\lambda)$ and the bare transparency $T_t(\lambda)$. Equation 5.61 essentially states that the amount of light absorption depends directly on the amount of absorbing material within the colorant, which in turn is proportional to both the concentration and thickness of the colorant layer. Often, the colorant thickness w is assumed to be spatially constant and is folded into the absorption coefficient $k(\lambda)$. To this end, we no longer explicitly include w in the analysis.

It is useful to introduce spectral transmittance $T(\lambda)$ and optical density $D(\lambda)$ of a colorant layer.

$$T(\lambda) = \frac{I(\lambda)}{I_i(\lambda)} = T_t(\lambda)\exp(-\xi k(\lambda));$$

$$D(\lambda) = -\log_{10}(T(\lambda)) = D_t(\lambda) + 0.4343\ \xi\ k(\lambda) \tag{5.62}$$

For color mixtures, the additivity rule can be invoked, which states that the density of a colorant mixture is equal to the sum of the densities of the individual colorants.[50] For a CMY printer, we thus have

$$D_{CMY}(\lambda) = D_t(\lambda) + 0.4343(\xi_C k_C(\lambda) + \xi_M k_M(\lambda) + \xi_Y k_Y(\lambda)) \tag{5.63}$$

which can be written in terms of transmittance,

$$T_{CMY}(\lambda) = T_t(\lambda) \exp[-(\xi_C k_C(\lambda) + \xi_M k_M(\lambda) + \xi_Y k_Y(\lambda))] \tag{5.64}$$

The model can be extended to reflective prints under the assumption that there is no scattering of light within the paper. This yields

$$R_{CMY}(\lambda) = R_p(\lambda) \exp[-(\xi_C k_C(\lambda) + \xi_M k_M(\lambda) + \xi_Y k_Y(\lambda))] \tag{5.65}$$

where $R_{CMY}(\lambda)$ and $R_p(\lambda)$ are the spectral reflectances of the colorant mixture and paper, respectively. Note that, in reality, most reflective media do exhibit scattering, hence reducing the accuracy of Equation 5.65.

For a given printer, the parameters of Equations 5.64 and 5.65 are estimated from measurements of selected patches. The procedure for a CMY reflection printer is as follows:

- Measure the spectral reflectance of the paper, $R_p(\lambda)$, and solid C, M, Y patches, $R_C(\lambda)$, $R_M(\lambda)$, $R_Y(\lambda)$.
- Estimate the spectral absorption coefficient $k_C(\lambda)$ for the cyan colorant. This is done by setting $\xi_C = 1$, $\xi_M = \xi_Y = 0$ in Equation 5.65 to yield

$$k_C(\lambda) = -\log\left(\frac{R_C(\lambda)}{R_p(\lambda)}\right) \tag{5.66}$$

  Use analogous expressions to derive $k_M(\lambda)$ and $k_Y(\lambda)$.

- Derive the relationship between input digital level d_j and cyan concentration ξ_{Cj} by printing a stepwedge of pure cyan patches at different digital levels d_j, and measure spectral reflectances $R_{Cj}(\lambda)$. From Equation 5.65, we know that

$$-\log\left(\frac{R_{Cj}(\lambda)}{R_p(\lambda)}\right) = \xi_{Cj} k_C(\lambda) \tag{5.67}$$

  The quantity on the left is the absorption corresponding to concentration at level d_j; hence, we denote this as $k_{Cj}(\lambda)$. A least-squares estimate for ξ_{Cj} can be computed by minimizing the error.

$$\sum_{\lambda} \left\| k_{Cj}(\lambda) - \xi_{Cj} k_C(\lambda) \right\|^2 \tag{5.68}$$

  where the summation is overall measured wavelengths within the visible spectrum. Using the least-squares analysis in Appendix 5.A, the optimal ξ_{Cj} is given by

$$\xi_{Cj}^{opt} = \frac{\sum_{\lambda} k_{Cj}(\lambda) k_C(\lambda)}{\sum_{\lambda} k_C^2(\lambda)} \tag{5.69}$$

By definition, these estimates lie between 0 and 1. Using Equation 5.69, we obtain a set of pairs (d_j, ξ_{Cj}), from which one-dimensional fitting or interpolation is used to generate a TRC that maps digital count to dye concentration for all digital inputs $0 \le d \le d_{max}$. This process is repeated for magenta and yellow.

This completes the model derivation process, and all the parameters in Equation 5.65 are known. The model can be tested by exercising it with an independent set of CMY test data. The model predictions are compared with actual measurements using a standard error metric such as ΔE_{94}. For efficiency of computation, the model can be used to create a three-dimensional LUT that maps CMY directly to CIE coordinates.

The BB model works very well for photographic transparencies and, to a reasonable extent, for photographic reflection prints. One of the shortcomings of this model is that it does not account for scattering within colorant layers, thus reducing its applicability for certain printing technologies. The scattering phenomenon is explicitly introduced in the Kubelka–Munk model, described next.

5.10.2.2 Kubelka–Munk model

The Kubelka–Munk (KM) model is a general theory for predicting the reflectance of translucent colorants. An appealing aspect of this theory is that it also models transparent and opaque colorants as special cases. The foremost applicability for printer characterization is the case of continuous-tone printing processes on reflective media. In this section, only the important formulae are presented. Their derivations are rather lengthy and can be found in many sources, including Allen.[51]

Kubelka–Munk theory assumes a special case of Figure 5.32b, with light being transmitted or scattered in only two directions: up and down. The most general form of the KM model for translucent colorant layers is given by

$$R(\lambda) = \frac{\frac{R_p(\lambda) - R_\infty(\lambda)}{R_\infty(\lambda)} - R_\infty(\lambda)\left(R_p(\lambda) - \frac{1}{R_\infty(\lambda)}\right)\exp\left[wS(\lambda)\left(\frac{1}{R_\infty(\lambda)} - R_\infty(\lambda)\right)\right]}{R_p(\lambda) - R_\infty(\lambda) - \left(R_p(\lambda) - \frac{1}{R_\infty(\lambda)}\right)\exp\left[wS(\lambda)\left(\frac{1}{R_\infty(\lambda)} - R_\infty(\lambda)\right)\right]} \tag{5.70}$$

where $R(\lambda)$ = the spectral reflectance of the sample
$R_p(\lambda)$ = the reflectance of the paper
w = the thickness of the colorant layer
$K(\lambda)$ and $S(\lambda)$ = absorbing and scattering coefficients, respectively

$R_\infty(\lambda)$ = the reflectance of an infinitely thick sample, given by

$$R_\infty(\lambda) = 1 + \frac{K(\lambda)}{S(\lambda)} - \sqrt{\left(\frac{K(\lambda)}{S(\lambda)}\right)^2 + 2\left(\frac{K(\lambda)}{S(\lambda)}\right)} \tag{5.71}$$

In practice, a sample is "infinitely thick" if any increase in thickness results in a negligible change in reflectance. Equation 5.71 can be inverted to obtain

$$\frac{K(\lambda)}{S(\lambda)} = \frac{(1 - R_\infty(\lambda))^2}{2R_\infty(\lambda)} \tag{5.72}$$

For colorant mixtures, the additivity and proportionality rules can be applied to obtain overall absorbing and scattering coefficients from those of the individual colorants.

$$K(\lambda) = k_p(\lambda) + \sum_{i = C,M,Y} \xi_i k_i(\lambda); \quad S(\lambda) = s_p(\lambda) + \sum_{i = C,M,Y} \xi_i s_i(\lambda) \tag{5.73}$$

where $k_p(\lambda)$ and $s_p(\lambda)$ = the absorption and scattering terms for the paper
ξ_i = the concentration of colorant i

The general KM model, Equation 5.70, can be simplified to the two limiting cases of transparent and opaque colorants (Figure 5.32a and 5.32c), described next.

5.10.2.2.1 KM model for transparent colorants. For transparent colorant layers, the scattering term in Equation 5.70 approaches zero, resulting in the following expression:[51]

$$R(\lambda) = R_p(\lambda)\exp[-2wK(\lambda)] \tag{5.74}$$

where $K(\lambda)$ is given by Equation 5.73. Note that this is very similar to the Beer–Bouguer model, Equation 5.65. However, the absorption coefficients in the two models are different, because BB assumes collimated light, whereas KM assumes diffuse light. The procedure outlined in Section 5.10.2.1 for the BB model can be used to estimate $k_i(\lambda)$ from C, M, Y samples at maximum concentration and to derive the mapping between input digital value d_i and dye concentration ξ_i from stepwedge measurements.

Berns has used this model to characterize dye diffusion printers.[52] In this work, the model parameters [i.e., $k_i(\lambda)$ and ξ_i, i = C, M, Y] were initially derived using essentially the procedure outlined in Section 5.10.2.1. A third-order polynomial was used to fit the relationship between digital count and dye concentration. The model resulted in unsatisfactory results (ΔE_{ab} = 12).

It was discovered that a major source of error arose from the channel independence assumption in the KM model, i.e., the cyan dye concentration depends only on the cyan digital count, etc. The author observed that, due to the sequential nature of the dye diffusion and transfer process, there is a significant sequential interaction among the colorants. This was accounted for by introducing a matrix with cross terms to relate KM predictions to more realistic estimates. Coefficients of the matrix were obtained by regression on a set of measurements of colorant mixtures. This correction was found to significantly improve the model prediction, resulting in $\Delta E_{ab} = 3$. Details are given in the Berns reference. The empirical correction just described is a common way of accounting for limitations in a physics-based model and will be encountered again in discussions of the Neugebauer model.

5.10.2.2.2 KM model for opaque colorants. For opaque samples, the limiting case of infinite thickness in Equation 5.71 can be used to predict spectral reflectance. Note that Equation 5.71 depends only on the ratio $K(\lambda)/S(\lambda)$ for the colorant mixture, which can be obtained from the absorption and scattering coefficients of the individual colorants using Equation 5.73.

$$\frac{K(\lambda)}{S(\lambda)} = \frac{k_p(\lambda) + \sum_{i=C,M,Y} \xi_i k_i(\lambda)}{s_p(\lambda) + \sum_{i=C,M,Y} \xi_i s_i(\lambda)} \tag{5.75}$$

This is referred to as the two-constant KM model. With certain pigments, it is reasonable to assume that the scattering in the colorants is negligible compared to scattering in the substrate.[51] In this case, the denominator in Equation 5.75 reduces to $s_p(\lambda)$, and Equation 5.75 can be rewritten as

$$\frac{K(\lambda)}{S(\lambda)} = \frac{k_p(\lambda)}{s_p(\lambda)} + \sum_{i=C,M,Y} \xi_i \frac{k_i(\lambda)}{s_i(\lambda)} \tag{5.76}$$

This is referred to as the single-constant KM model, as only a single ratio $k(\lambda)/s(\lambda)$ is needed for each colorant.

To derive the model, the $k(\lambda)/s(\lambda)$ terms for each colorant are obtained from reflectance measurements of samples printed at maximum concentration, using Equation 5.72. Next, the relationship between digital count and colorant concentration ξ are obtained from reflectance measurements of single-colorant stepwedges and a regression procedure similar to that outlined in Section 5.10.2.1. Finally, Equations 5.76 and 5.71 are evaluated in turn to obtain the predicted reflectance. More details are found in papers by Parton et al.[53] and Kang.[54] In these papers, the opaque single-constant KM model is used to predict the spectral reflectance of solid area coverage in inkjet prints. The prediction accuracies are in the range of 1.65 to 5.7 ΔE^*_{ab}

depending on the ink mixing process and the particular mixtures tested. Note that most inkjet printers use halftoning, a process that is not well predicted by KM theory. The latter only predicts the solid overprints in inkjet prints, hence its application is in ink formulation rather than device characterization.

5.10.2.2.3 Modeling front-surface and interlayer reflections. An important effect not taken into account in the KM and BB models is reflection loss at the boundaries between colorant layers, as well as front surface reflection (FSR) at the boundary between the uppermost colorant layer and air. Because a certain amount of light is lost due to FSR, this should ideally be subtracted before computing reflectance. However, in a spectrophotometer, at least part of the light from FSR reaches the detector. To correct for this effect, Saunderson[55] developed a relationship between the reflectance $R(\lambda)$ as predicted by BB or KM, and the reflectance $R_{meas}(\lambda)$ as measured by a spectrophotometer.

$$R(\lambda) = \frac{R_{meas}(\lambda) - k_1}{1 - k_1 - k_2(1 - R_{meas}(\lambda))}; \; R_{meas}(\lambda) = k_1 + \frac{(1 - k_1)(1 - k_2)R(\lambda)}{1 - k_2 R(\lambda)} \tag{5.77}$$

where k_1 is the Fresnel reflection coefficient that accounts for front surface reflection, and k_2 models total internal reflection that traps light within the colorant layers. The factor k_1 depends on the refractive index η of the uppermost colorant layer. A common assumption for η is 1.5, which corresponds to $k_1 = 0.04$. The theoretical value of k_2 for the case of perfectly diffuse light is 0.6.[51] Alternatively, these parameters can be chosen to provide the best empirical fit between measured and modeled reflectance data.

The Saunderson correction is performed as a final step after deriving the BB or KM, and it has been shown to improve model accuracy.[52,54]

5.10.2.2.4 Modeling fluorescence. Another drawback with both the BB and KM models is that they do not account for fluorescence. Many paper substrates employ optical brighteners that exhibit fluorescence and can thus limit the utility of these models. Fluorescence modeling is discussed in more detail in Chapter 3, which deals with the physics of color.

5.10.2.3 Neugebauer model

The Neugebauer model is used to model a halftone color printing process. Each primary colorant in a halftone process is rendered as a spatial pattern of dots, each dot being printed at one of a small number of concentration levels. The impression of intermediate levels is achieved by modulating the size, shape, and spatial frequency of the dots. (Techniques for color halftoning are covered in more detail in a subsequent chapter.)

A process employing N colorants at Q concentration levels results in one of Q^N colorant combinations being printed at any given spatial location. We begin the formulation with the simplest case of a binary black-and-white

printer. This corresponds to $N = 1$ and $Q = 2$ (zero or maximum) concentration levels; thus, at any given spatial location, we have two possible colorant combinations, black or white. The reflectance of a halftone pattern is predicted by the Murray–Davies equation,[50]

$$R = (1 - k)P_p + kP_k \tag{5.78}$$

where k = fractional area covered by the black dots
P_p, P_k = reflectances of paper and black colorant, respectively

The Neugebauer model is a straightforward extension of the Murray–Davies equation to color halftone mixtures.[56] Binary printers employing C, M, Y colorants render one of $2^3 = 8$ colorant combinations at a given spatial location. The set of colorant combinations is S = {P, C, M, Y, CM, MY, CY, CMY}, where P denotes paper white, C denotes solid cyan, CM denotes the cyan–magenta overprint, etc. The Neugebauer model predicts the reflectance of a color halftone as a weighted average of the reflectances of the eight colorant combinations.

$$R = \sum_{i=S} w_i P_i \tag{5.79}$$

where S = the aforementioned set of colorant combinations
P_i = spectral reflectance of the ith colorant combination, henceforth referred to as the ith Neugebauer primary
weight w_i = the relative area coverage of the ith colorant combination, which is dictated by the halftoning method used

In the original Neugebauer equations, the predicted color is specified by three broadband reflectances representing the short, medium, and long wavelength portions of the electromagnetic spectrum. In this work, spectrally narrowband reflectances are used instead of their broadband counterparts, as the former generally yield greater accuracy.[57] The spectral Neugebauer equations are

$$R(\lambda) = \sum_{i \in S} w_i P_i(\lambda) \tag{5.80}$$

Because $P_i(\lambda)$ are colors of solid overprints, they can be predicted from single-colorant measurements using the BB or KM theories described in the previous sections. However, for any given set of colorants, there are only a small number of such overprints; hence, they are usually measured directly.

5.10.2.3.1 Effect of halftone dot placement. A common assumption is that the dot placements of the colorants are statistically independent; i.e.,

the event that a particular colorant is placed at a particular spatial location is independent of other colorants being placed at the same location. This leads to the Demichel dot model.[50] The Neugebauer primaries and the corresponding weights are given by

$$\begin{aligned} P_i(\lambda) \in S_P &= \{P_P(\lambda), P_C(\lambda), P_M\lambda, P_Y(\lambda), P_{CM}(\lambda), P_{CY}(\lambda), P_{MY}(\lambda), P_{CMY}(\lambda)\}, \\ w_i \in S_w &= \{(1-c)(1-m)(1-y), c(1-m)(1-y), m(1-c)(1-y), \\ &\quad y(1-c)(1-m), cm(1-y), cy(1-m), my(1-c), cmy\} \end{aligned} \tag{5.81}$$

Here, c, m, y are the fractional area coverages corresponding to digital inputs d_c, d_m, d_y, respectively. A halftone screen for which statistical independence is often assumed is the rotated halftone screen configuration, where the screens for c, m, y are placed at different angles, carefully selected to avoid moiré artifacts. This is shown schematically in Figure 5.33a. Validity of the independence assumption for certain types of halftones such as rotated screens has been demonstrated by Viggiano et al.[58]

A geometrical interpretation of Equations 5.80 and 5.81 is that $R(\lambda)$ is a result of trilinear interpolation performed among the $P_i(\lambda)$ in cmy space. (This can be verified by comparing these equations with the trilinear interpolation equations given in Chapter 11, dealing with efficient color transformations.) An algebraic interpretation of the model is that Equations 5.80 and 5.81 form a third-order polynomial in terms of c, m, y, with $P_i(\lambda)$ being the polynomial coefficients.

Another commonly used halftone configuration is the dot-on-dot screen,[59] where the C, M, Y dots are placed at the same screen angle and

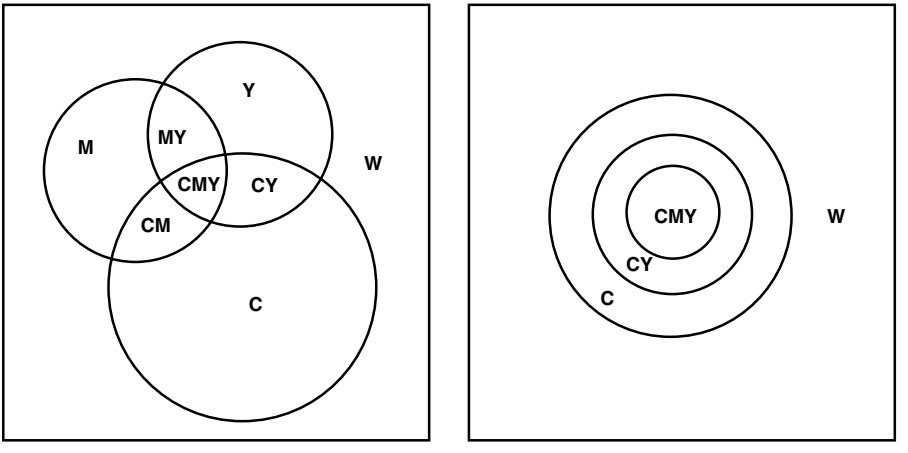


Figure 5.33 Dot area coverages for (a) rotated screen and (b) dot-on-dot screen.

phase as shown in Figure 5.33b. While the basic form of the mixing equations is similar to Equation 5.80, the weights w_i are different from those of randomly positioned dots. Let X_i be the colorant with the ith smallest area coverage a_i. For example, if [c, m, y] = [0.8, 0.5, 0.2], then X_1 = Y, X_2 = M, X_3 = C; $a_1 = 0.2$, $a_2 = 0.5$, $a_3 = 0.8$. The set of Neugebauer primaries and corresponding weights are now given by

$$P_i(\lambda) \in S_P = \{P_{X_1X_2X_3}(\lambda),\ P_{X_2X_3}(\lambda),\ P_{X_3}(\lambda),\ P_w(\lambda)\},$$
$$w_i \in S_w = \{a_1, a_2 - a_1, a_3 - a_2, 1 - a_3\} \tag{5.82}$$

The final output reflectance in Equation 5.80 is now a summation of, at most, four terms.

Geometrically, Equation 5.82 represents tetrahedral interpolation among the $P_i(\lambda)$ at four of the eight vertices of the *cmy* cube. (This can be verified by comparing Equation 5.82 with the equations for tetrahedral interpolation given in the Chapter 11.) Different Neugebauer primaries are selected for the calculation depending on the relative sizes of the area coverages *c, m, y* (equivalently, the tetrahedron to which the input *cmy* coordinate belongs). However, the weights w_i, and hence the resulting interpolated output $R(\lambda)$, are continuous as the input *cmy* coordinate moves from one tetrahedron to another in *cmy* space. Algebraically, it is easily seen from Equations 5.80 and 5.82 that, for fixed λ, $R(\lambda)$ is a linear function of *c, m, y*, with the $P_i(\lambda)$ being the weighting coefficients.

The aforementioned dot-on-dot mixing model assumes an ideal dot pattern with no noise, a perfectly rectangular dot density profile, and no misregistration effects. In practice, these assumptions may be violated. It has been shown[59] that a weighted mixture of the dot-on-dot and Demichel mixing models can effectively capture some of these effects. The new predicted reflectance is given by

$$R'(\lambda) = (1 - \alpha)R_{dod}(\lambda) + \alpha R_{dem}(\lambda) \tag{5.83}$$

where $R_{dod}(\lambda)$ = reflectance predicted by the dot-on-dot model
$R_{dem}(\lambda)$ = reflectance predicted by the Demichel model
α = a weighting parameter that determines the relative proportions of the two mixing models; this factor can be chosen to fit the model to a set of measured data

As alluded to earlier, all these versions of the Neugebauer model easily generalize for an arbitrary number of colorants. For N colorants, the Demichel model for independent dot placement will result in the summation in Equation 5.80 containing 2^N terms, while the dot-on-dot model contains $N + 1$ terms.

5.10.2.3.2 Effect of halftone screen frequency. The ideal Neugebauer model assumes a perfect rectangular dot profile as a function of spatial location. In reality, dots have soft transitions from regions with full colorant to regions with no colorant. If the halftone screen frequency is relatively low, or a clustered dot is used, the relative area of the paper covered by the transition regions is small, and the Neugebauer model would be expected to be relatively accurate. On the other hand, if the screen frequency is high, or a dispersed dot is used, a relatively large fraction of the paper is covered by transition regions, and the model breaks down. While some of the corrections discussed in following sections partially account for soft transitions, the reliability of the model has been seen to be greatest with clustered dot screens with frequency less than 100 halftone dots per inch.

5.10.2.3.3 Effect of light scattering in the paper. An important phenomenon not modeled by the basic Neugebauer equations is the scattering of light within the paper. To understand this phenomenon, consider the interaction of light with a black halftone print. The light that reaches the paper is given by

$$I_p = I_{in}(1 - k + kT_k) \tag{5.84}$$

where I_{in} = incident light intensity
I_p = light reaching the paper
k = fractional black area coverage
T_k = transmittance of the black colorant

Figure 5. 34a shows the case where there is no optical scattering within the paper. In this case, light incident on the print at a location containing colorant will also exit through the colorant; likewise, light reaching the substrate will exit from the same location. The reflected light is thus given by

$$I_{refl} = I_{in}\{(1 - k)\ P_p + k\ T_k^2\ P_p\} \tag{5.85}$$

where P_p is reflectance of the paper.

Define the reflectance of the solid black colorant as $P_k = T_k^2 P_p$. The overall reflectance is then given by the original Murray–Davies Equation 5.78.

Consider now the case where there is scattering within the paper, as shown in Figure 5.34b. In this case, light that enters the paper through an area with no colorant may leave the paper through an area that is covered with colorant, and vice versa. To account for this, Yule and Nielsen[60] proposed a simple correction to the Murray–Davies model for a black printer. Assuming that light reaching the paper is given by I_p in Equation 5.84, the light emerging from the substrate is $I_p R_p$. If complete diffuse scattering is assumed, the light is equally likely to re-emerge from the paper in all directions. In this case,

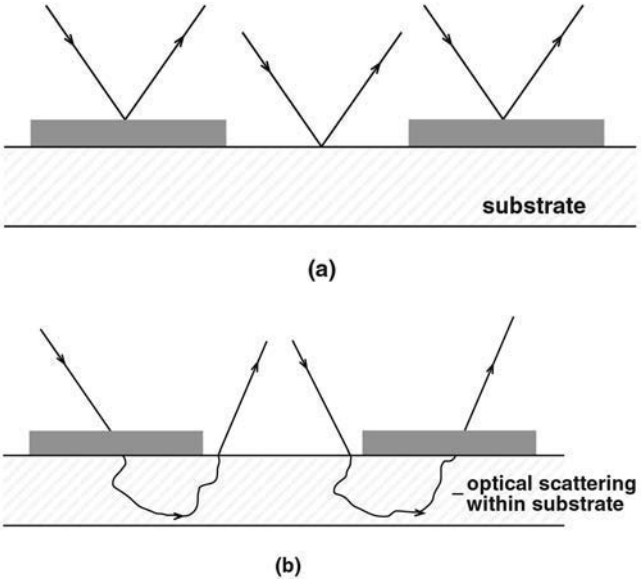


Figure 5.34 Light reflection (a) without and (b) with optical scattering within the substrate.

the emerging light experiences the same transmission function, $(1 - k + kT_k)$, as in Equation 5.84. The final reflected light is given by

$$I_{reft} = I_p P_p (1 - k + kT_K) = I_{in} P_p (1 - k + kT_K)^2 \tag{5.86}$$

With the black reflectance being defined as $P_k = T_k^2 P_p$, the following expression is obtained for the overall reflectance:

$$R = \frac{I_{reft}}{I_{in}} = ((1 - k)P_p^{1/2} + kP_k^{1/2})^2 \tag{5.87}$$

The Yule–Nielsen (YN) correction results in a nonlinear relationship between the area coverage k and the resulting reflectance R. Figure 5.35 is a plot of R vs. k with and without the YN correction. The latter predicts a smaller reflectance (i.e., a darker print) than the linear Murray–Davies model. This is indeed the case in reality. The darker print can be thought of as being effected by a larger dot area coverage k; hence, the scattering phenomenon is often referred to as *optical dot gain.*

Equation 5.87 can be generalized as follows:

$$R = ((1 - k)P_p^{1/n} + kP_k^{1/n})^n \tag{5.88}$$

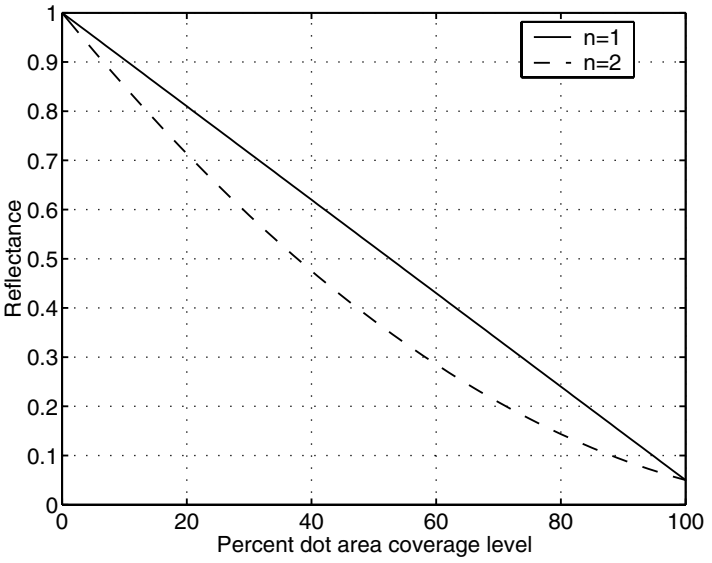


Figure 5.35 Reflectance vs. area coverage for *K* colorant (a) without Yule–Nielsen correction ($n = 1$) and (b) with Yule–Nielsen correction ($n = 2$).

where n is known as the YN parameter. When $n = 1$, Equation 5.88 reduces to the Murray–Davies equation, i.e., the case of no optical scattering within the substrate. When $n = 2$, we have the case of complete diffuse optical scattering given in Equation 5.87. In reality, one would expect to encounter partial diffuse scattering, which would yield intermediate values, $1 < n < 2$. Therefore, n is often treated as a free parameter chosen to optimally fit measured data. The YN correction is readily applied to the spectral Neugebauer equations.

$$R(\lambda) = \left(\sum_{i \in S} w_i P_i(\lambda)^{1/n} \right)^n \tag{5.89}$$

Figure 5.36 is a plot of the prediction accuracy of the Neugebauer model as a function of n for a rotated screen. The device being modeled was a Xerox 5760 CMYK laser printer. Details of the experiment that produced these results are given in the paper by Balasubramanian.[59] Clearly, inclusion of the YN factor (i.e., $n > 1$) greatly improves model accuracy. Interestingly, for this case, best results are achieved for $n > 2$, for which there is no direct physical interpretation. Statistical or empirical fitting of model parameters can indeed often result in nonphysical values. This is largely due to noise and other characteristics such as front surface and internal reflections not being sufficiently captured by the given model.

Other, more sophisticated techniques have been proposed that model optical scattering with spatial point spread functions.[61,62] These approaches

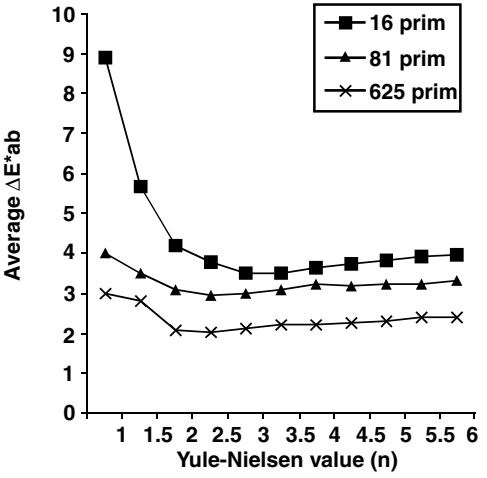


Figure 5.36 Average ΔE vs. YN parameter n for spectral Neugebauer model with $2^4 = 16$, $3^4 = 81$, and $5^4 = 625$ primaries, for rotated dot screen.

are covered in more detail in Chapter 3. The following discussion is restricted to the YN correction, as it is a very simple yet effective way of improving model accuracy.

5.10.2.3.4 Estimation of dot area coverages. In addition to the optical dot gain just described, halftone printing also experiences mechanical dot gain, which results from the physical spreading of colorant on the paper. A combination of optical and mechanical dot gain results in a nonlinear relationship between the input digital counts to the halftone function and the dot area coverages used in the Neugebauer calculation. Furthermore, in some printing processes, optical interactions among the colorants can result in the dot gain for a given colorant being dependent on the area coverages of the other colorants. However, for pedagogical purposes, we will make the simplifying assumption that there are no interchannel interactions so that the cyan area coverage depends on only the cyan digital count, etc. This assumption is reasonably upheld in many printing processes and allows the relationship between digital count and dot area to be determined from single-colorant stepwedge data. From Equation 5.89, the reflectance of a cyan patch produced at digital level d_j is given by

$$R_{C_i}(\lambda)^{1/n} = (1 - c_j)P_p(\lambda)^{1/n} + c_j P_C(\lambda)^{1/n} \tag{5.90}$$

The least-squares estimate minimizes the error

$$E = \sum_{\lambda \in V} [R(\lambda)_{C_j}^{1/n} - ((1 - c_j)P_P(\lambda)^{1/n} + c_j P_C(\lambda)^{1/n})]^2 \tag{5.91}$$

The optimal area coverage is obtained by setting to zero the partial derivative of Equation 5.91 with respect to c_j, yielding

$$c_j^{opt} = \frac{\sum_{\lambda}(P_p(\lambda)^{1/n} - R(\lambda_{c_j})^{1/n})(P_P(\lambda)^{1/n} - P_C(\lambda)^{1/n})}{\sum_{\lambda}(P_P(\lambda)^{1/n} - P_C(\lambda)^{1/n})^2} \tag{5.92}$$

The result is a set of pairs $\{d_j, c_j\}$ from which a continuous function can be derived that maps digital count to dot area coverage using some form of one-dimensional fitting or interpolation. The process is repeated for the other colorants. If a sufficiently fine sampling of stepwedge data is available, piecewise linear interpolation should be adequate; otherwise, higher-order functions such as splines are desirable. Figure 5.37 shows optimized magenta dot areas for the DocuColor 12 printer for values of $n = 1$ and $n = 2$. For the case where $n = 1$, the dot area coverages must account entirely for both optical and mechanical dot gain. When $n > 1$, the YN correction partially accounts for optical dot gain; hence, the dot area coverages are generally smaller in magnitude.

An alternative technique for determining dot areas is to minimize the error in CIELAB rather than spectral coordinates. Unlike the previous approach, this is a nonlinear optimization problem that must be solved with numerical or search-based techniques. Given this fact, one can extend the training set to include colorant mixtures, e.g., C = M = Y, in addition to the single-colorant stepwedges. Balasubramanian[59] provides further details of this approach.

5.10.2.3.5 Cellular Neugebauer model. The set of primaries $P_i(\lambda)$ of the basic Neugebauer model are derived from C, M, Y overprints of either 0 or 100% area coverages. This set can be generalized to include intermediate

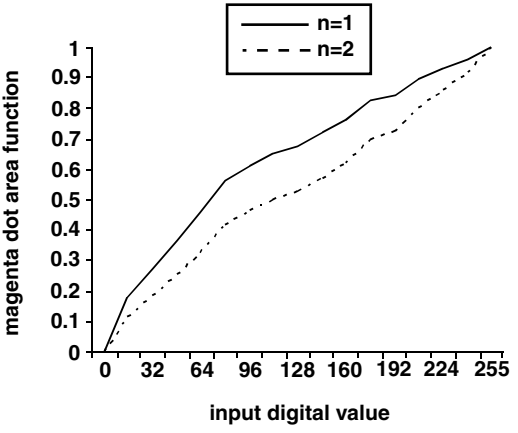


Figure 5.37 Optimized magenta dot area functions for $n = 1$ and 2.

area coverages. For example, if 50% area coverages of C, M, Y are included with 0 and 100%, then each colorant has three states, and there are $3^3 = 27$ Neugebauer primaries. Geometrically, this is equivalent to partitioning the three-dimensional *cmy* space into a grid of eight rectangular cells, formed by nodes at 0, 50, and 100%. Hence, this is referred to as the cellular Neugebauer model.[63] A two-dimensional example is shown in Figure 5.38 for a printer employing only cyan and magenta colorants. Depending on the type of halftone screen, the appropriate mixing equations are applied within each cell. The mixing equations are to be geometrically interpreted as a three-dimensional interpolation of the $P_i(\lambda)^{1/n}$ at the cell vertices. For the case of the random halftone, the logical extension from the noncellular model is to perform trilinear interpolation within each cell whereas, for the dot-on-dot case, tetrahedral interpolation is to be applied.

More explicitly, a given set of dot areas c, m, y can be represented as a point in three-dimensional *cmy* space and will fall in a rectangular cell that is bounded by the lower and upper extrema, denoted c_l, c_u, m_l, m_u, y_l, y_u, along each of the three axes. That is, c_l and c_u are the two points along the cyan axis that satisfy the constraint $0 \le c_l < c < c_u \le 1$; c_l, $c_u \in I_c$, where I_c is the set of allowable states or area coverages for the Neugebauer primaries corresponding to the cyan colorant. Analogous definitions hold for the magenta and yellow coordinates. To estimate the reflectance within a given cell, the dot area values c, m, y, must be normalized to occupy the interval [0, 1] within that cell.

$$c' = \frac{c - c_l}{c_u - c_l} \tag{5.93}$$

with analogous expressions for m' and y'. The weights w_i' for the cellular model are then given by Equation 5.81 for random screens and Equation 5.82 for dot-on-dot screens, with c, m, y being replaced by c', m', y', respectively. Let $P_i'(\lambda)$ be the spectral Neugebauer primaries that correspond to the vertices of the enclosing cell. The mixing equations for the cellular model are then given by Equation 5.89, with w_i replaced by w_i' and P_i replaced by P_i'.

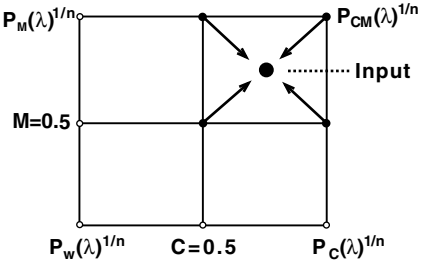


Figure 5.38 Two-dimensional illustration of cellular Neugebauer model. Solid circles denote spectral primaries interpolated to obtain reflectance $R(\lambda)^{1/n}$ at the input *cm* value.

Note that the cellular equations physically model a halftoning process wherein each colorant can produce $M > 2$ concentration levels. For binary printers, the justification for using a cellular model is empirical rather than physical; the finer cellular subdivision of *cmy* space affords finer interpolation of measured data, hence yielding greater accuracy.

Figure 5.36 compares the accuracy of the noncellular model for a CMYK printer with cellular versions employing $3^4 = 81$ and $5^4 = 625$ primaries. As the number of cells increases, the model accuracy improves significantly. At the same time, the dependence on the YN factor decreases. This is to be expected, as the cellular model marks a transition from a model-based to an empirical approach and hence would be less sensitive to model parameters.

5.10.2.3.6 Spectral regression of the Neugebauer primaries. Thus far, the primaries $P_i(\lambda)$ in Equation 5.89 are considered as fixed parameters that are directly measured. An alternative is to treat these quantities as free variables that can be optimized via regression on a training set of spectral reflectance data. This technique will be described next for the case of a noncellular CMY model employing rotated screens. (Extension to the cellular case, N colorants, or dot-on-dot screen is straightforward). It is assumed that the optimal n factor and dot area coverages have been derived using the aforementioned techniques. To formulate the regression problem, it is convenient to express the Neugebauer equations in matrix-vector form. Consider each spectral measurement as an L-vector. Collect the YN modified spectral reflectances $R(\lambda)^{1/n}$ of T training samples into a $T \times L$ matrix **R**. Similarly, collect the YN modified Neugebauer primaries $P_i(\lambda)^{1/n}$ into an $8 \times L$ matrix **P**. Finally, generate a $T \times 8$ weight matrix **W** whose element w_{ij} is the area coverage of the jth Neugebauer primary for the ith training sample. Equation 5.89 can then be rewritten as

$$\mathbf{R} = \mathbf{W} \cdot \mathbf{P} \tag{5.94}$$

From Appendix 5.A, the least squares solution for **P** is given by

$$\mathbf{P}_{opt} = (\mathbf{W}^t\mathbf{W}^{-1})\mathbf{W}^t\mathbf{R} \tag{5.95}$$

The terms in **P** are raised to the power n to obtain optimized primary reflectances. It must be emphasized that the choice of CMY samples in the training set **T** is crucial in determining the condition or rank of matrix **W**. Namely, to ensure sufficient rank, the samples should be chosen so that there are no null columns in **W**. A simple way to assure this is to pick a regular three-dimensional grid of training samples. Also, note that the foregoing analysis is based on a particular choice of n and the dot area coverage functions. The process can be iteratively repeated by rederiving n and the dot areas corresponding to the newly optimized primaries, and then repeating the regression step. Experiments by the author have shown that more than two iterations do not generally yield significant improvements in model accuracy.[59]

5.10.2.3.7 Overall model optimization. The following procedure may be used to optimize the various parameters of the Neugebauer model for a CMY printer:

- Select the resolution of the cellular Neugebauer model. In the author's experience, three levels (i.e., two cells) per colorant offers an acceptable trade-off between accuracy and number of samples required. Generate CMY combinations corresponding to the cell nodes (i.e., the Neugebauer primaries).
- Select the resolution of C, M, Y stepwedges to generate dot area functions. In the author's experience, a minimum of 16 samples per colorant is usually adequate.
- Select an additional set of CMY mixtures to test and refine the model. One possibility is to use an $N \times N \times N$ grid of CMY combinations that does not coincide with the Neugebauer primaries.
- Combine the above CMY samples into a characterization target. (As an alternative to designing a custom target, the standard IT8.7/3 printer characterization target described in Section 5.3 can be used, as it contains the patches necessary to derive and test the Neugebauer model.) Print the target and obtain spectral measurements.
- For a fixed value of n (e.g., $n = 1$), use Equation 5.92 to generate estimates of dot area coverages for the stepwedge samples. Interpolate or fit the data to create functions that map digital count to dot area coverages. With 16 or more samples per stepwedge, piecewise linear interpolation should produce adequate accuracy.
- Evaluate the accuracy of the model in predicting the stepwedge data. This is accomplished by computing a ΔE metric between model predictions with actual measurements.
- Optimize the model with respect to n by repeating the previous two steps for several n values in some nominal range (e.g., $1 < n < 7$) and selecting the n that produces the minimum ΔE.
- Select a mixing model depending on the type of halftone screen (e.g., Demichel vs. dot-on-dot).
- If the dot-on-dot model is chosen, find the optimal blending parameter α in Equation 5.83 by iterating through different values of α, computing ΔE for the model's prediction of mixed color samples from the test set, and selecting α that minimizes the ΔE.
- If spectral regression of the primaries is desired, select a set of mixed color samples from the test set, and use Equation 5.95 to compute optimal primaries $P_i(\lambda)$.

Figure 5.39 summarizes the steps in the application of the Neugebauer model. Accuracy of the model must be evaluated on an independent set of CMY samples. If the prediction error is within the variability of the printer, the model is considered to be a satisfactory representation of the real printer.

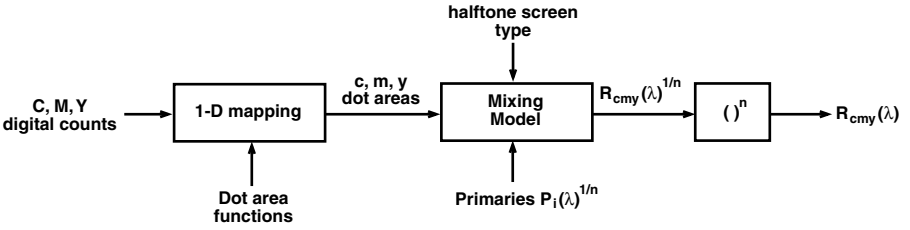


Figure 5.39 Block diagram of Neugebauer model calculation.

5.10.2.3.8 Accuracy of the various Neugebauer models. Table 5.1 compares the performance of the various types of Neugebauer models applied to the Xerox 5760 CMYK printer. Details are provided by Balasubramanian.[59] Clearly, the YN parameter offers significant benefit to the model. The cellular framework with $5^4 = 625$ primaries offers the best accuracy, but this is at the expense of a substantial number of measurements. The cellular model with $3^4 = 81$ primaries, as well as spectral regression, offer a promising trade-off between measurement cost and accuracy.

Table 5.1 Effort Involved and Resulting Accuracy of the Various Neugebauer Models for a Rotated Dot

Model	No. of Spectral Measurements	Avg. ΔE^*_{ab}	95% ΔE^*_{ab}
Basic spectral	72	8.88	16.3
Yule–Nielsen corrected	72	3.50	7.80
Cellular, 3^4 primaries, Yule–Nielsen corrected	137	2.96	6.0
Cellular, 5^4 primaries, Yule–Nielsen corrected	681	2.01	5.0
Yule–Nielsen corrected, global spectral regression	188	2.27	5.3

5.10.2.3.9 Further enhancements. Several researchers have explored other refinements of the model. Arney et al.[64] showed that the colors of both the paper and the dots are functions of the relative dot area coverages, and they extended the Neugebauer model to account for this. Lee et al.[65] departed from the Demichel model and used a sequential quadratic programming method to estimate these parameters. Iino and Berns[66] accounted for optical interactions among the colorants by introducing a correction to the dot gain of a given colorant that depends on the area coverages of the other colorants. Hua and Huang[67] and Iino and Berns[68,69] explored the use of a wavelength-dependent Yule–Nielsen factor. Agar and Allebach[70] developed an iterative technique of selectively increasing the resolution of a cellular model in those

regions where prediction errors are high. Xia et al.[71] used a generalization of least squares, known as total least-squares (TLS) regression to optimize model parameters. Unlike least-squares regression, which assumes uncertainty only in the output space of the function being approximated, total least-squares assumes uncertainty in both the input and output spaces and can provide more robust and realistic estimates. In this regard, TLS has wide applicability in device characterization.

5.10.3 Empirical techniques for forward characterization

With this class of techniques, a target of known device-dependent samples is generated, printed, and measured, and the characterization function is derived via data fitting or interpolation. Linear regression is generally inadequate for printer characterization; any of the more sophisticated nonlinear techniques described in Section 5.4 are applicable.

5.10.3.1 Lattice-based techniques

Perhaps the most common approach is to generate a regular grid of training samples in m-dimensional device space, print and measure these samples, and use a lattice-based technique to interpolate among the measured colorimetric values (see Section 5.4.5). There is an inherent trade-off between the size and distribution of the sample set and the resulting accuracy. This trade-off must be optimized based on the particular printer characteristics and accuracy requirements. Remember that, if the printer has been calibrated, these functions must be incorporated into the image path when generating the target; hence, they will also affect the overall printer characteristics. If, for example, the printer has been calibrated to be linear in ΔE from paper along each of the primary axes (see Section 5.10.1), then uniform spacing of lattice points is a good choice, as these correspond approximately to equal visual steps. The following is a simple procedure to determine a suitable grid size for a CMY printer, assuming it has been either linearized channelwise to ΔE from paper or gray-balanced and linearized to neutral L^*:

- Generate uniformly spaced lattices of size s^3 in CMY space, where $5 \le s \le 10$. Also generate an independent test target of CMY samples. The latter can be generated by invoking a random number generator for each of the digital values d_c, d_m, d_y or by using a regular lattice that is different from any of the training sets.
- Generate targets for both the lattice and the test data, process through the calibration functions, print, and measure CIELAB values.
- From this data, generate a set of three-dimensional LUTs of size s^3 that map CMY to CIELAB space.
- Select a three-dimensional interpolation technique, e.g., trilinear or tetrahedral interpolation. Process the test CMY samples through each of the LUTs to obtain CIELAB estimates. Compute ΔE between estimated and measured CIELAB.

- Plot average and 95th percentile ΔE as a function of s. A logical choice for the lattice size is the smallest s for which an increase in lattice size does not yield appreciable reduction in 95th percentile ΔE value.

Figure 5.40 shows such a plot for a Xerox DocuColor 12 laser printer. This plot suggests that, for this printer, there is no appreciable gain in increasing the grid size beyond $s = 8$.

The extension to CMYK printers is straightforward. Note, however, that the lattice size (hence, the number of measurements) increases as s^4 and can quickly become prohibitively large. One method of improving the trade-off between lattice size and accuracy is sequential interpolation, described next.

5.10.3.2 *Sequential interpolation*

The general framework for sequential interpolation (SI) was introduced in Section 5.4.6. Here, we describe a specific application to CMYK characterization. Consider a decomposition of CMYK space into a family of CMY subspaces corresponding to different levels of K, as shown in Figure 5.41. If we were to print and measure the CMYK nodes of each CMY lattice, we would obtain a series of volumes in $L^*a^*b^*$ space, as illustrated schematically in the same figure. Each gamut volume corresponds to variations in C, M, and Y, with fixed K. Note that as K increases, the variation in color, and hence the gamut volume, decreases. For the case where K = 100%, we have almost negligible color variation. The fact that the curvature of the function strongly depends on K motivates an SI structure comprising a family of CMY

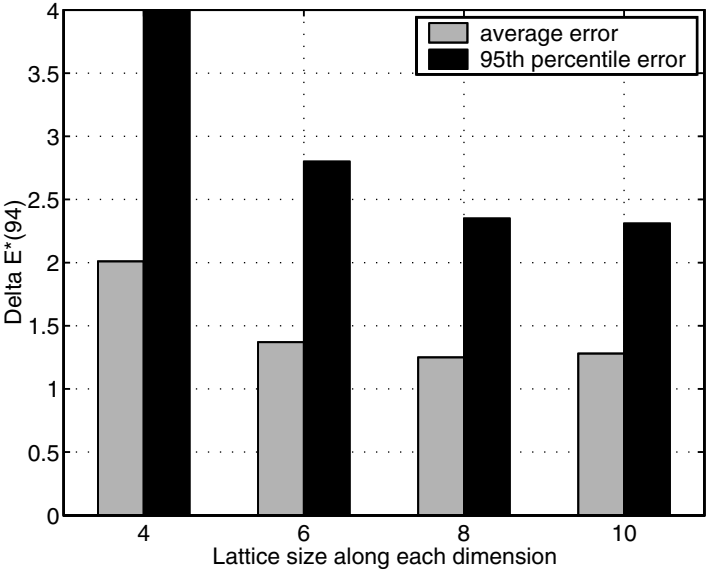


Figure 5.40 ΔE vs. lattice size.

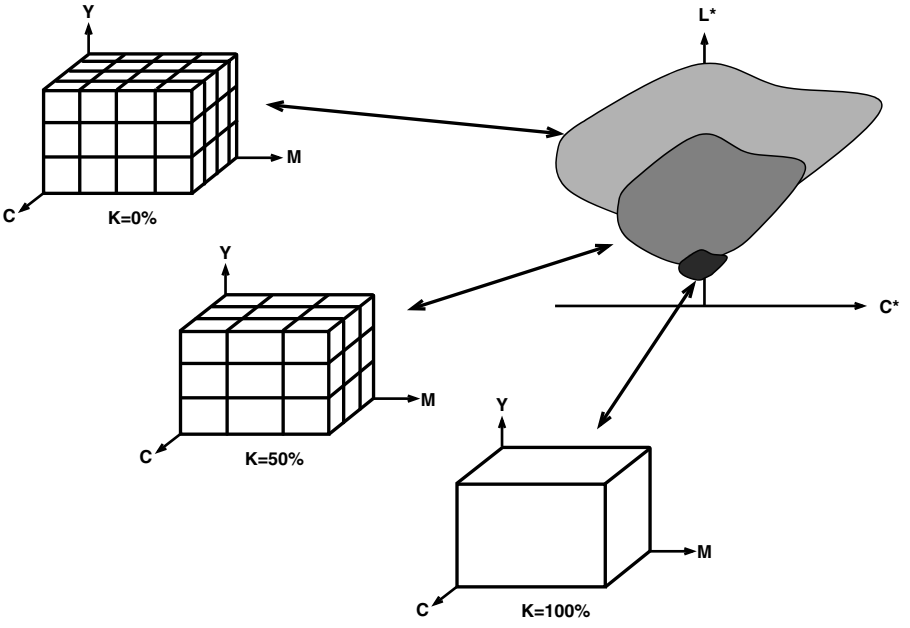


Figure 5.41 Sequential interpolation: a decomposition of CMYK into a family of CMY subspaces at different K and corresponding CIELAB gamuts. The CMY lattices become coarser as K increases.

lattices for different K. A finely sampled CMY lattice is used for K = 0, and the lattice size decreases with increasing K, as shown in Figure 5.41. When building the SI structure, each CMY lattice is filled with measured CIELAB values. Interpolation to map CMYK to CIELAB is performed as follows:

- Project the input CMYK point onto the K dimension and select neighboring levels K_j and K_{j+1}.
- Project the input CMYK point onto CMY space and perform three-dimensional interpolation on the two CMY lattices corresponding to levels K_j and K_{j+1} to produce two CIELAB points.
- Use the input K value to perform one-dimensional interpolation of these two CIELAB points.

Table 5.2 shows experimental results comparing the SI structure with a regular lattice. For approximately the same lattice size, the SI technique offers superior accuracy, hence improving the quality/cost trade-off. Further details are given by Balasubramanian.[72]

It is noteworthy that the standard IT8.7/3 printer characterization target described in Section 5.3 facilitates SI. The target contains 6 CMY lattices of size 6^3, 6^3, 5^3, 5^3, 4^3, 2^3, corresponding to K values (in percentage) of 0, 20, 40, 60, 80, 100, respectively.

Table 5.2 Comparison of Accuracy and Number of Training Samples for Standard vs. Sequential Interpolation

Model	CIE '94 ΔE		Number of LUT Nodes
	Average	95th Percentile	
Regular 4 × 4 × 4 × 4 lattice	3.0	12.3	256
Sequential interpolation with 5^3, 4^3, 3^3, 2^3 CMY lattices corresponding to k = 0, 85, 170, 255	1.8	6.25	224

5.10.3.3 *Other empirical approaches*

Tominaga (Chapter 9 of Reference 7) describes an example of a neural network for printer characterization. This is accomplished in two steps. First, a four-layer neural net is derived for the forward transform from CMYK to CIELAB using over 6500 training samples. Next, a cascaded eight-layer neural net is constructed, the first stage being the inverse mapping from CIELAB to CMYK and the second stage being the previously derived forward mapping from CMYK to CIELAB. The second stage is kept static, and the first stage is optimized to minimize the CIELAB-to-CIELAB error for the overall system. Tominaga reports an average ΔE_{ab} of 2.24 for a dye sublimation printer. As with the other techniques, the optimal number of training samples and the neural net structure depend on the printer characteristics and desired accuracy, and they have to be initially determined by trial and error.

Herzog[73] proposes an analytic model for the mapping between CMY and CIELAB. The printer gamut is described as a family of nested shells in both CMY and CIELAB space. A simple mathematical model of distortion and scaling operations is used to relate each shell from one space to another via an intermediate representation called a *kernel gamut*. Colors in between the shells are mapped via linear interpolation. A total of 626 measurements are required to derive the model, and average ΔE_{ab} errors between 0.7 and 2.5 are reported for various data sets.

5.10.4 *Hybrid approaches*

We have seen thus far that physical models and empirical techniques offer different trade-offs between effort and accuracy. There are two ways to combine the strengths of these two classes of techniques. The first is to use empirical data to optimize the parameters of a physics-based model. Many examples of this were encountered in the optimization of BB, KM, and Neugebauer models. The second is to use empirical data to refine the prediction of a printer model as a post-processing step, as shown in Figure 5.42. The assumption is that the model is a good first-order approximation, and that a small number of additional refinement samples is sufficient to correct for objectionable inaccuracies in the model.[12] The number and distribution

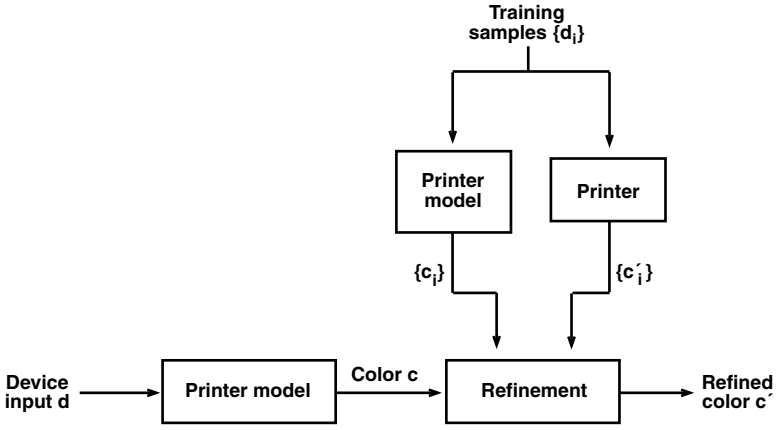


Figure 5.42 Block diagram showing refinement of printer model.

of refinement samples depend on the characteristics of the printer and the model, as well as on accuracy requirements. If the printer model is known to be erroneous in certain regions of color space, the refinement samples can be chosen with a denser sampling in these regions. Similarly, regions of color space to which the human visual system is more sensitive (e.g., flesh tones and neutral colors) can be sampled more densely. In the absence of such information, a reasonable approach is to span the gamut with an approximately uniform sampling.

In the case of forward printer characterization, the refinement is a colorimetric function from, for example, CIELAB to CIELAB. Any of the multidimensional data-fitting or interpolation techniques described in Section 5.4 can be applied to estimate this function from the refinement samples. Local linear regression has been used successfully by the author[12] to reduce average ΔE_{ab} errors from approximately 5 to 2.5.

5.10.5 *Deriving the inverse characterization function*

The inverse printer characterization is a mapping from CIE color to device colorant values that, when rendered, will produce the requested CIE color under defined viewing conditions. This mapping is usually implemented as a three-dimensional LUT, so it needs to be evaluated at nodes on a regular three-dimensional lattice in CIE coordinates. Some of the lattice nodes will lie outside the printer gamut; we assume that these points are first mapped to the gamut surface with a gamut-mapping step (described in Chapter 10). Hence, we restrict the inversion process to colors that are within the printer gamut.

In the case where the forward function is described by an analytic model, a possible approach is to directly invert the parameters of the model via analytic or search-based techniques. The most notable efforts in this direction

have been in the inversion of the Neugebauer model to estimate dot area coverages from colorimetric values.[74,75] Here, we adopt a more general inversion process that is independent of the technique for determining the forward function. The process is accomplished in two steps.

1. Use the forward characterization function to generate a distribution of training samples $\{\mathbf{c}_i, \mathbf{d}_i\}$ in device-independent and device-dependent coordinates.
2. Derive the inverse function by interpolating or fitting this data.

5.10.5.1 *CMY printers*

In the case of three-colorant devices, the forward function from CMY to colorimetric coordinates (e.g., CIELAB) is a unique mapping; hence, a unique inverse exists. Any of the interpolation or fitting techniques described in Section 5.4 can be used to determine the inverse function from the training samples. Tetrahedral inversion, described in Section 5.4.5, can be used if the device signals are generated on a regular lattice. Figure 5.43 compares four fitting algorithms (local linear regression, neural network, polynomial regression, and tetrahedral inversion) as to their ability to invert a Neugebauer model derived for a Xerox DocuColor12 laser printer. The neural network

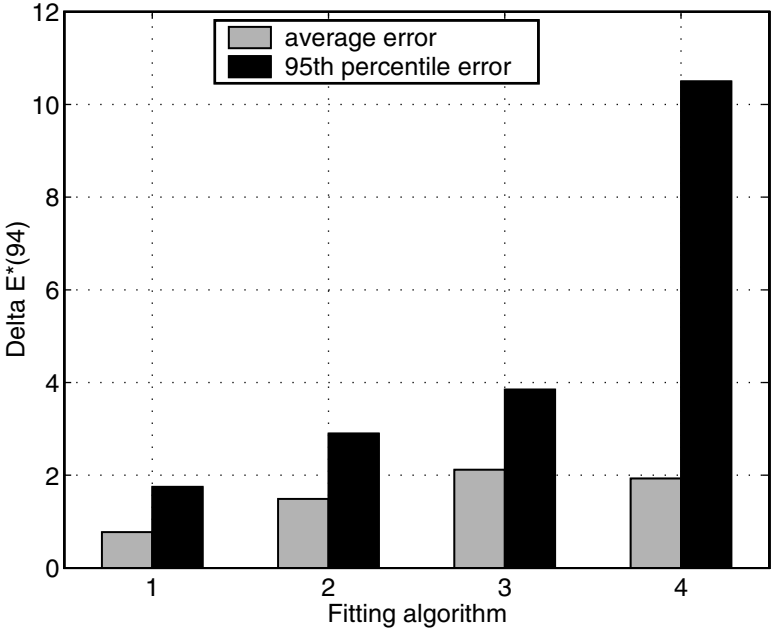


Figure 5.43 Comparison of various algorithms used to invert a Neugebauer model: 1. local linear regression, 2. neural network, 3. polynomial regression, and 4. tetrahedral inversion.

used a feed-forward algorithm with one hidden layer containing six neurons. The polynomial regression used a 3×11 matrix as in Equation 5.18b. A training set of $10^3 = 1000$ samples was used to derive the parameters for each of the fitting algorithms. An independent set of 125 samples was used as the test set. The test data, specified in CIELAB, were mapped through a given inverse algorithm to obtain CMY, which was then mapped through the forward printer model to obtain reproduced CIELAB values. The plot in Figure 5.43 shows the average and 95% ΔE^*_{94} errors between the original and reproduced values. Local linear regression and the neural network offer the best performance. In the author's experience, this observation holds generally true for a wide variety of printers. Local linear regression possesses the added advantage that it is less computationally intensive than the neural network.

Another factor that affects the overall inversion accuracy is the size of the three-dimensional LUT used to finally approximate the inverse function. An experiment was conducted to study overall inversion error as a function of LUT size. The workflow is the same as described in the preceding paragraph, except that the inverse function is now a three-dimensional LUT built using local linear regression on 1000 training samples. Figure 5.44 is a plot of overall inversion error as a function of LUT size. The error decreases with increasing LUT size; however, beyond a certain point, the returns diminish. From the plot, it is clear that a LUT size beyond $16 \times 16 \times 16$ does not afford a noticeable gain in accuracy — another observation that has been seen to

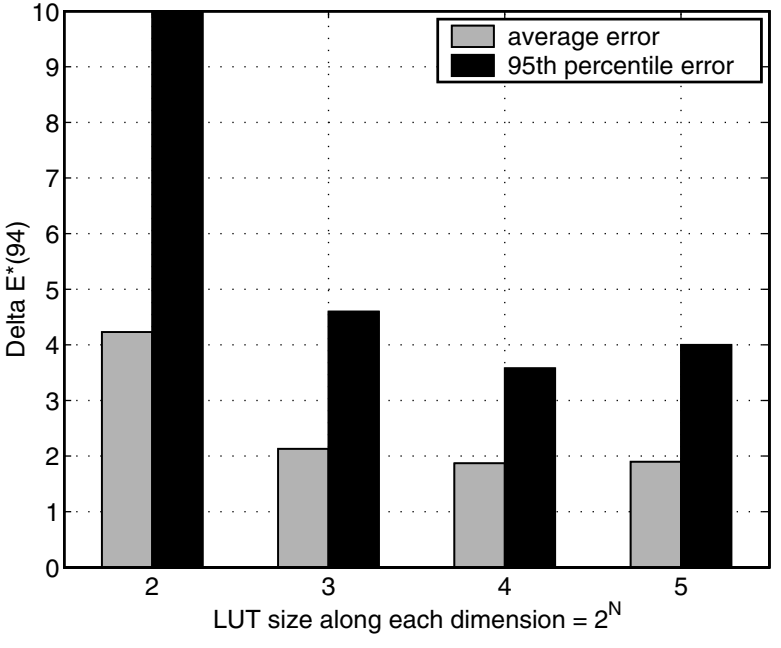


Figure 5.44 LUT approximation error vs. LUT size.

hold true for a wide variety of printers. Note that the relative spacing of nodes along each dimension can also affect LUT accuracy. In this experiment, the nodes were spaced uniformly, because the input space, CIELAB, in which the LUT was built, is approximately visually uniform.

5.10.5.2 CMYK printers

Although, in principle, the three C, M, and Y colorants suffice to produce all perceivable hues, very often, a fourth black (K) colorant is used for several reasons. First, the K colorant is usually considerably less expensive than C, M, and Y, and it can thus be used in lieu of CMY mixtures to render dark neutrals and shadows. Second, the addition of K can result in an increase in gamut in the dark regions of color space in comparison to what is achievable using only CMY mixtures. Third, the use of K can help reduce the total amount of colorant required to produce a given color, a feature that is critical in certain technologies such as inkjet printing.

In the context of device characterization, the K colorant introduces redundancy into the forward transform, as a large (in principle, infinite) number of CMYK combinations can result in the same colorimetric measurement. This results in the inverse function being ill posed, and additional constraints are required to generate a unique CMYK combination for each input CIE color. Some common methods of deriving the constrained inverse are presented next.

5.10.5.2.1 Inversion based on K addition, undercolor removal, and gray component replacement. The processes of black (K) addition, undercolor removal (UCR), and gray component replacement (GCR) trace their origins to the graphic arts printing industry.[50] Together, they define a unique transform from a set of canonical CMY primaries to the CMYK signals for the given printer. Geometrically, the transform generates a three-dimensional manifold within the four-dimensional CMYK space, with the property that every CMYK combination within the manifold results in a unique colorimetric response. Once this transform is established, the inversion can be carried out on the canonical CMY device as described in Section 5.10.5.1. Figure 5.45 shows the derivation and application of the inverse function for a CMYK printer. The two functions in Figure 5.45b are usually concatenated into a composite inverse transform from CIE to CMYK signals. Recall that the printer is assumed to have been calibrated, so the CMYK signals resulting form the inversion process are finally processed through the calibration functions prior to printing. In some implementations, the calibration is concatenated with the characterization or stored in the same profile.

There are numerous methods for designing K addition, UCR, and GCR functions. They are usually chosen for an optimal trade-off among factors such as gamut volume, colorant area coverage, and smoothness of transitions from neutral to non-neutral colors. The trade-off is usually carried out

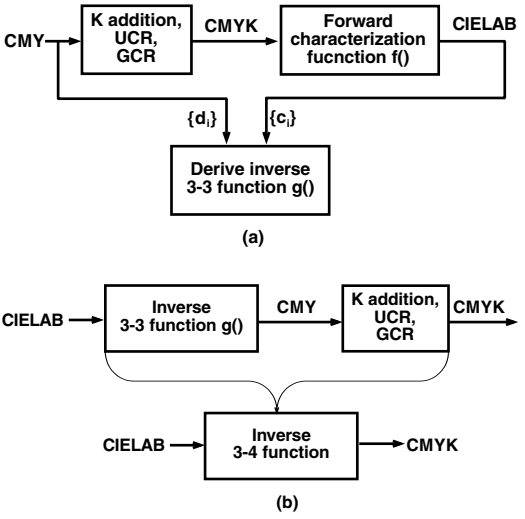


Figure 5.45 Constrained inverse characterization of CMYK printers: (a) construction of inverse 3–3 function and (b) combining the inverse 3–3 function with K addition, UCR, and GCR to construct inverse 3–4 function.

heuristically with knowledge of the printer characteristics and quality requirements. Some examples of these functions are presented next.

5.10.5.2.1.1 Black addition. This is commonly chosen to meet a desired behavior along the C = M = Y axis. Suppose the printer has been gray-balanced and linearized to neutral L^*. If we define darkness D^* as a scaled inverse of L^* using Equation 5.58, then we have C = M = Y = D^* along the neutral axis for the range $0 \le D^* \le D^*_{max}$. Here, D^*_{max} is the maximum digital count (e.g., 255 for an 8-bit system). We can then define K as a monotonic increasing function f_1 of neutral D^*. Numerous functional representations can be used, for example the power-law,

$$f_1(D^*) = \begin{cases} D^*_{max}\left(\dfrac{D^* - D^*_{offset}}{D^*_{max} - D^*_{offset}}\right)^{\gamma} & \text{if } D^*_{offset} < D^* \le D^*_{max} \\ 0 & \text{if } (0 \le D^* \le D^*_{offset}) \end{cases} \tag{5.96}$$

Here, γ and D^*_{offset} are parameters that can be adjusted to suit the desired behavior of K along the neutral axis. For γ > 1, larger values of γ and D^*_{offset} result in less aggressive f_1 (i.e., less K is used for a given amount of neutral C = M = Y). As γ and D^*_{offset} approach 1 and 0, respectively, f_1 becomes more aggressive, with the amount of K approaching the amount of neutral C = M = Y.

5.10.5.2.1.2 Undercolor removal. This function describes the amount of reduction in CMY primaries to compensate for the K addition. It is also derived with attention to the neutral axis. A simple form of CMY reduction is given by

$$C' = C - f_2(D^*) \tag{5.97}$$

with analogous expressions for M and Y. Again, we are abounded with numerous strategies for $f_2(D^*)$. One approach is based on the rationale that the CMY reduction should be proportional to the amount of K addition,

$$f_2(D^*) = \alpha f_1(D^*), \quad 0 \le \alpha \le 1 \tag{5.98}$$

The case where $\alpha = 1$ (i.e., CMY subtraction equals K addition) is often referred to as 100% UCR.

A more sophisticated approach is to derive f_2 to colorimetrically compensate for the K addition. This can be performed as follows. For a given neutral input C = M = Y sample, the resulting L^* and hence D^* that would be produced by printing this sample can be predicted via the forward characterization function. The amount of K associated with this input C = M = Y is given by $f_1()$. We can now derive the new smaller amounts, C′ = M′ = Y′, which produce the same D^* when combined with the given K. This step is achieved by combining different C = M = Y levels with the given K, running through the forward transform, and picking the combination that produces the desired D^*. Finally, f_2 is the difference between the original C = M = Y and final C′= M′= Y′.

A key factor to be considered in choosing black addition and UCR parameters is the total area coverage (TAC) that is permissible for the given device, especially in the dark portions of the gamut. For many CMYK printers, TACs near 400% will result in defects (e.g., ink bleeding in inkjet printers or improper toner fusing and flaking in xerographic printers). Hence, an appropriate limit must be placed on TAC, and this in turn affects the K addition and UCR parameters. If the colorimetric approach described in the preceding paragraph is adopted, accuracy will likely be sacrificed toward the dark end of the gamut due to TAC limits.

5.10.5.2.1.3 Gray component replacement. Thus far, K addition and UCR have been defined for neutral samples C = M = Y. GCR is a generalization of these functions for the entire gamut of CMY combinations. A fundamental assumption is that the gray component of an arbitrary CMY combination is given by the minimum of C, M, Y. This gray component can then be used as input to the K addition and UCR functions.

$$X = \min(C,M,Y)$$

$$K = f_1(X)$$

$$C' = C - f_2(X)$$

$$M' = M - f_2(X)$$

$$Y' = Y - f_2(X) \tag{5.99}$$

Clearly, one can conceive numerous enhancements to this simple model. The CMY subtraction can be performed in other spaces such as optical density. This can be accomplished in the current framework by applying a transform to the chosen space before CMY subtraction and applying the inverse transform after subtraction. Second, functions f_1 and f_2 can be multidimensional functions that depend on more than just the minimum of C, M, Y. This may be desirable if, for example, the optimal balance between K and CMY along the neutral axis is different from that along the edges of the gamut. Finally, CMY reduction can be accomplished by methods other than simple subtraction. For another perspective on UCR and GCR techniques, refer to Holub et al.[76]

5.10.5.2.2 Direct constraint-based CMYK inversion. The previous approach used a CMY-to-CMYK transform to arrive at a constrained inverse. A more general and direct method is to obtain the set of all CMYK combinations that result in the given input CIE color and select a combination that satisfies certain constraints. Examples of such constraints include:

1. Total colorant area coverage (i.e., C + M + Y + K) is less than a threshold.
2. The amount of K with respect to the minimum and maximum K that can produce the given color is constrained.
3. Stability is maximized (i.e., change in colorant values results in minimum change in CIE color).
4. Smoothness is maintained with respect to neighboring colors in CIE space.
5. Gamut volume is maximized.
6. Spatial artifacts such as misregistration and moiré are minimized or constrained.

Constraint 5 implies that, if a color is achievable with only one CMYK combination, this should be used, even if some of the other constraints are not met. Space considerations do not permit us to elaborate on the other constraints. We refer the reader to Mahy[77] for detailed discussions of constraints 1 through 4), and Balasubramanian et al.[78] for a description of how UCR and GCR are optimized to minimize moiré. Cholewo[79] describes another constrained inversion technique that takes gamut mapping into account.

Note that the aforementioned two approaches can be combined. For example, the UCR/GCR approach can be used to generate an initial CMYK combination, which can then be refined iteratively to meet one or more of constraints 1 through 6.

5.10.6 Scanner-based printer characterization

All the foregoing discussion implicitly assumes the use of a colorimeter or spectrophotometer for color measurement. Another approach is to use a color scanner as the measurement device in printer characterization. The main advantage of this approach is that scanning a color target is less labor intensive than spectrophotometric measurement; furthermore, the measurement time does not depend on the number of patches on the target. Because a scanner is generally not colorimetric, it must first be characterized. Fortunately, the characterization needs to be derived only for the colorant-medium combination used by the given printer, hence empirical techniques can be used with very accurate results (see Section 5.6.3). The scanner characterization produces a transform from scanner RGB to colorimetric or spectral coordinates, thus turning the scanner into a colorimetric device for the given colorants and medium. Printer characterization can then be carried out as described in the foregoing sections, with the target measurement step being replaced by scanning of the target followed by the necessary image processing (i.e., mapping the scanned image through the scanner characterization and extracting colorimetric values of each patch). Note that this approach intimately links the scanner and printer into a characterized pair.

5.10.7 Hi-fidelity color printing

The term *high-fidelity (hi-fi) color* refers to the use of extra colorants in addition to the standard C, M, Y, K. Two strategies can be adopted for hi-fi color printing. In one approach, the additional colorants are of the same hues as the standard colorants but of different concentrations. Usually, C and M are chosen for this. The use of multiple concentrations allows for superior rendering of detail in the highlights and shadows as well as smoother transitions from highlights to mid-tones to shadows. In the second strategy, the extra colorants are of hues that are different from C, M, Y. One purpose of this is to extend the color gamut. Due to unwanted absorptions, printer gamuts are often deficient in saturation and lightness in the secondary overprints, i.e., R, G, B. Hence, hi-fi colorants are chosen to extend the gamut in these regions of color space, as shown in the schematic in Figure 5.46. Another purpose is to reduce metamerism and enable spectral reproduction by using colorants with relatively narrowband spectral characteristics (see Section 5.11).

5.10.7.1 Forward characterization

For forward characterization, the BB, KM, and Neugebauer models all extend in a straightforward manner to an arbitrary number of colorants. In the case of BB and KM, the number of measurements increases linearly with the number of colorants while, for the Neugebauer model, the number of measurements increases exponentially due to the need for including solid overprints. In the latter case, the number of overprints can become prohibitively large; hence, a two-stage model can be employed. The BB or KM is used to predict

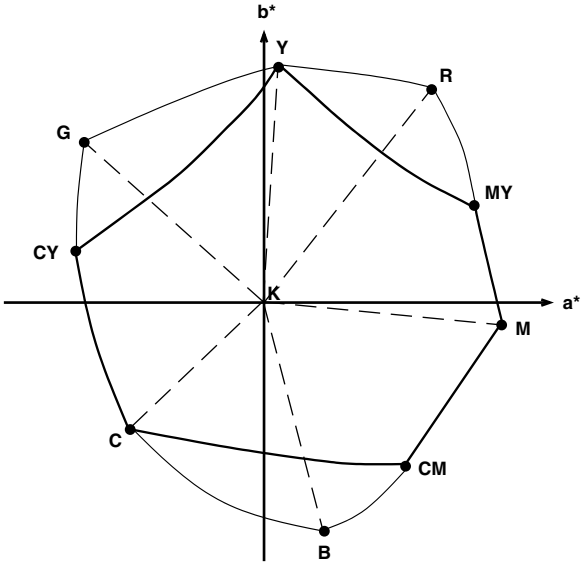


Figure 5.46 Use of R, G, B hi-fi colorants to extend the gamut achieved with standard CMYK printing.

the reflectances of solid overprints, and this is fed to the Neugebauer model, which predicts the reflectances of arbitrary colorant mixtures. The reader is referred to Section 6.6 of Reference 7 for further discussions and references.

Recently, a novel hi-fi color modeling approach has been proposed by Van de Capelle and Meireson (Reference 7, Chapter 10) that obviates the need for measuring overprints even for halftone processes. The main advantage is therefore the substantial savings in number of measurements as the number of colorants increases. Single-colorant stepwedges between 0 and 100% area coverage are printed under three conditions:

1. On the naked substrate
2. On the substrate with 50% black
3. On the substrate with 100% black

From these measurements, three substrate-independent parameters are estimated for each colorant, namely scattering, interaction, and absorption. These parameters describe spectral properties of the colorants independent of the underlying substrate and can be used to predict the reflectance of a colorant layer on any given substrate. The reflectance of an n-layer mixture of colorants is modeled iteratively by calculating the reflectance of the $n - 1$ layer mixture and treating this as the substrate for the nth layer. The authors report average ΔE_{ab} between 1.8 and 3.0 for various data sets. See the aforementioned reference for details.

5.10.7.2 *Inverse characterization*

As with CMYK printing, hi-fi color introduces redundancy into the color reproduction process in that many colorant combinations can result in the same perceived (CIE) color. The inversion process must select a unique colorant combination for each input CIE color. Additional considerations include minimization of colorant area coverage, minimization of moiré in the case of rotated halftone screening, and a smooth characterization function from CIE to device signals.

A common strategy is to partition the color gamut into subgamuts formed from combinations of three or four colorants. Referring to Figure 5.46, one approach is to partition the gamut into six subgamuts formed from the following colorant combinations: CGK, GYK, YRK, RMK, MBK, and BCK. Here, R, G, and B represent hi-fi colorants chosen in the red, green, and blue hues.[80,81] (Often, orange is used instead of red; the idea still holds.) Because C + Y combinations are spectrally redundant with G, the former are disallowed, and only C + G and G + Y combinations are entertained. This rule guarantees uniqueness, and is applied likewise for the other combinations. Also, because only three screens angles are being used to render any given input color, moiré can be handled using the same rules applied for conventional printing. Another variant[82] represents the gamut as overlapping subgamuts formed by GYRK, YRMK, RMBK, MBCK, BCGK, and CGYK. Several criteria, including minimization of colorant area coverage and smoothness of transitions across subgamuts, are used to select unique colorant combinations for a given input color. A third variant[83] employs an approach directly akin to UCR and GCR for CMYK printing. The gamut is partitioned into the subgamuts formed by YRMK, MBCK, and CGYK. In the YRMK subgamut, the M and Y signals are fed to an R addition function and MY subtraction function. Analogous operations are applied in the other subgamuts. The functions are chosen to produce a unique colorant combination for each input color while maximizing the volume of each subgamut. As with conventional UCR/GCR, the latter can be aided by applying a nonlinearity to the input signal and inverting this after the addition and subtraction operations have been performed. Note that, with all these approaches, hi-fi colorants are used to render colors that can also be achieved with standard CMYK colorants. In a fourth variant, only CMYK colorants are used to achieve colors within the standard CMYK gamut, and hi-fi colorants are introduced only in those regions of color space that cannot be reproduced with CMYK mixtures. This approach offers better compatibility with standard CMYK devices. Details of these techniques are deferred to the stated references.

5.10.8 *Projection transparency printing*

Overhead transparency projection continues to be a common medium for communication of color information. Characterization for this application is complicated by several factors. First, the final viewing conditions and

projector characteristics are difficult to predict *a priori* and may be very different from those used for characterization. Second, it is difficult to achieve strict spatial uniformity, especially when the image is projected into a large screen. Fortunately, color accuracy requirements for projected transparency are usually not so stringent as to require careful characterization. But, if accuracy is important, care must be taken to control the aforementioned conditions as well as possible.

The characterization procedure is conceptually the same as for conventional printers. A test target is printed on a transparency and projected under representative viewing conditions. The main factor that affects the latter is ambient lighting. The projected image is measured with a spectroradiometer. To minimize light scattering, the transparency can be masked to present only one patch at a time to the spectroradiometer. The measured CIELAB values and original printer coordinates are then used to build the characterization transform. For all the reasons stated above, a simple transparency model such as the Beer–Bouguer formula in Equation 5.64 may not suffice to predict the characterization; rather, empirical approaches and three-dimensional LUTs may be necessary to capture the various complex effects. The techniques described in Section 5.4 and Section 5.10.3 can be used for this purpose. For further reading, see Chapter 10 of Kang[48] and the work by Cui et al.[84]

5.11 Characterization for multispectral imaging

The discussion in this chapter is primarily based on colorimetric reproduction wherein the device-independent representation of Figure 5.1 comprises three color channels. An emerging area of research that is gaining increasing attention is multispectral imaging, whereby the goal is to capture, store, and reproduce narrowband spectral information rather than three-dimensional colorimetric data. A primary motivation for preserving spectral information is that it mitigates the metamerism problems encountered in conventional colorimetric reproduction. A related advantage is that colorimetric characterizations can be computed dynamically for different viewing illuminants from one set of spectral data. A schematic of a spectral imaging system is shown in Figure 5.47. The input device records color in N channels, where N is greater than three. This is usually accomplished by using N narrowband filters in the image acquisition device. Spectral information is reconstructed from the data via spectral characterization of the input device.

A major concern in spectral imaging is the substantial increase in the amount of data to be handled (i.e., from 3 to 30 or more channels). This necessitates an efficient encoding scheme for spectral data. Most encoding techniques are based on the well-known fact that spectra found in nature are generally smooth and can be well approximated by a small number (i.e., between five and eight) of basis functions.[85] The latter can be derived via principal-component analysis (PCA) described in an Chapter 1. PCA yields a compact encoding for spectral data and can serve as the device-independent color space in a multispectral imaging framework. An important con-

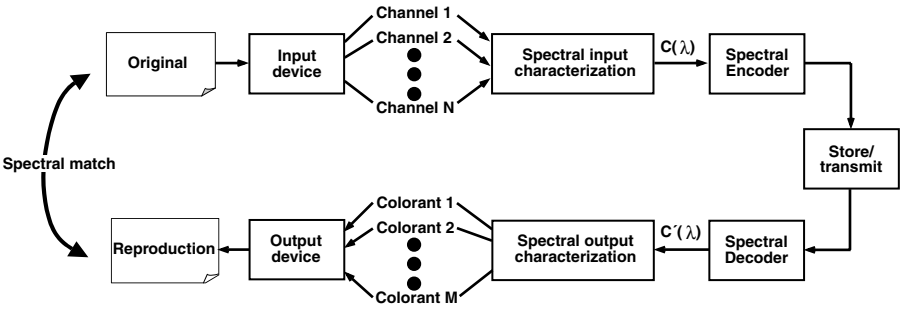


Figure 5.47 Multispectral imaging system.

sideration in selecting the PCA encoding is to ensure compatibility with current colorimetric models for color management.[86]

The goal of the output device is to reproduce the spectral (rather than colorimetric) description of the input image via a spectral characterization. As with input devices, a "spectral printer" must employ more than the traditional C, M, Y colorants to facilitate spectral reproduction. The forward characterization transform can be achieved using many of the techniques described in this chapter; derivation of the inverse transform is, however, a more challenging problem.

For further details, the reader is referred to the book by MacDonald and Luo[7] for a description of multispectral image capture and encoding techniques; the work by Tzeng and Berns[87–89] for contributions to multispectral printing, including colorant selection and characterization algorithms; and Rosen et al.[90] for a description of a framework for spectral characterization.

5.12 *Device emulation and proofing*

Frequently, it is desired to emulate the color characteristics of one device on another. Two classic examples are the use of a proof printer to emulate a color press and the use of a softcopy display to emulate a printer. (The latter is known as *softproofing*.) In both cases, the idea is to use a relatively inexpensive and easily accessed device to simulate the output of a device that is less accessible and for which image rendering is costly. We will generically refer to these two devices as the proofing device and target device, respectively. Device emulation is a cost-effective strategy when iterative color adjustments on an image are needed prior to final production. The iterations are carried out on the proofing device, and the final production takes place on the target device.

Figure 5.48 is a flow diagram showing how a device emulation transform can be generated from characterizations of the proofing and target devices. A device-independent color input $\mathbf{c}$ is transformed to the target device space $\mathbf{d}_t$ via the function $g_t()$. This transform should be the same as the one used for final rendering of images on the target device. Because the target device

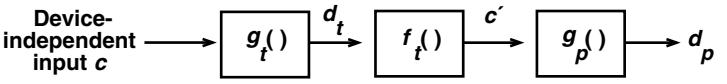


Figure 5.48 Block diagram for device emulation. Functions $f()$ and $g()$ denote forward and inverse characterizations. Subscripts "t" and "p" refer to target and proofing devices. Device-independent and device-dependent color representations are denoted $\boldsymbol{c}'$ and $\boldsymbol{d}$.

is invariably a printer, $\mathbf{d}_t$ is usually a CMYK representation. Next, the device colors are transformed back to device-independent coordinates $\mathbf{c}'$ via the target device's forward transform $f_t()$. Thus, $\mathbf{c}'$ describes the appearance of the given input color on the target device. The remaining step is to match this color on the proofing device. This is accomplished by applying the inverse transform $g_p()$ for the proofing device and rendering the resulting device coordinates $\mathbf{d}_p$ to this device. Depending on whether the proofing device is a printer or display, $\mathbf{d}_p$ will be in CMYK or RGB space, respectively. For efficiency, the operations in Figure 5.48 are usually concatenated into a single emulation transformation.

In an alternative scenario, the input image may already exist in the target device space $\mathbf{d}_t$, in which case only the last two blocks in Figure 5.48 need to be executed. A common example of this occurs when CMYK files prepared for a standard offset press (e.g., SWOP) are to be rendered on a digital CMYK proofing device. The result is a four-dimensional CMYK-to-CMYK transform.

The four-to-four transform deserves brief mention. We learned in Section 5.10.5.2 that constraints are needed to produce a unique output CMYK combination for each distinct input color, and we introduced the notion of K addition, UCR, and GCR for this purpose. In the case where the input is a CMYK space, an alternative constraint might be to determine the output K as a function of input K. The simplest instantiation is to simply preserve the input K signal. Techniques of this type are presented by Cholewo[91] and Zeng.[92]

5.13 Commercial packages

A number of calibration and characterization products are available that offer a wide range of capabilities and performance. Comprehensive product descriptions are beyond the scope of this chapter. However, for the more application-oriented reader, the following is a list of mainstream color management products available at the time this book was published.

Note that this list mainly includes stand-alone packages and that calibration and characterization functions are also often embedded within device controllers (e.g., print controller products by Creo-Scitex, Electronics for Imaging) or as applications bundled with current operating systems (e.g., Adobe Gamma for display calibration).

GretagMacbeth (ProfileMaker)	www.gretagmacbeth.com
Agfa (ColorTune)	www.agfa.com/software/colortune.html
ColorBlind (Matchbox)	www.color.com
ColorSavvy (WiziWYG Pro)	www.colorsavvy.com
Kodak (ColorFlow)	www.kodak.com
LinoColor (Scanopen, Viewopen, Printopen)	www.linocolor.com
Monaco Systems (MonacoEZcolor and MonacoProfiler)	www.monacosystems.com
Praxisoft (WiziWYG, CompassProfile)	www.praxisoft.com/products/cms.html

5.14 Conclusions

In this chapter, we hope to have provided the reader with the theoretical foundation as well as practical procedures and guidelines to accomplish device calibration and characterization. The chapter began with a general conceptual overview and terminology associated with color characterization of input and output devices. Two basic approaches to characterization were presented: model-based and empirical. In addition, hybrid techniques were described that combine the strengths of both approaches. Next, a treatment was given of fundamental topics that apply to all forms of device characterization — namely color measurement technology, data interpolation and fitting algorithms, and quantitative analysis tools. This was followed by a detailed discussion of the calibration and characterization of several common input and output devices, including scanners, digital cameras, displays, and printers. Finally, the chapter concluded with several special topics, namely characterization of hi-fi and projection transparency printers, device emulation techniques, and commercial color characterization products.

Clearly, there are many aspects to this subject, and we have not been able to cover all of them in great depth. It is hoped that the extensive set of references will serve for further enquiry into any given topic. Finally, it must be emphasized that device characterization is not a topic that stands on its own, and it cannot by itself guarantee high-quality results in a color imaging system. The latter calls for a thorough system-wide understanding of all the components in the color imaging chain and their interactions. This is evidenced by the numerous cross references to other chapters in this book.

Acknowledgment

The author wishes to thank Dean Harrington for his assistance in preparing the figures for this chapter.

References

1. *Postscript Language Reference Manual*, 2nd ed., Addison-Wesley, Reading, MA, Chap. 6.

2. Trussell, H. J., DSP solutions run the gamut for color systems, *IEEE Signal Processing*, 10, 8–23, 1993.
3. Luther, R., Aus Dem Gebiet der Farbreizmetrik, *Z. Tech. Phys.*, 8, 540–558, 1927.
4. Ives, H. E., The transformation of color-mixture equations from one system to another, *J. Franklin Inst.*, 16, 673–701, 1915.
5. Sharma, G. and Trussell, H. J., Color scanner performance trade-offs, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts*, J. Bares, Ed., Vol. 2658, 1996, 270–278.
6. Sharma, G. Trussell, G, H. J, and Vrhel, M. J., Optimal non-negative color scanning filters, *IEEE Trans. Image Proc.*, 7(1), 129–133, 1998.
7. MacDonald, L. W. and Luo, M. R., Eds., *Colour Imaging — Vision and Technology,* Wiley, Chichester, U.K., 1999.
8. Alessi, P. J. and Cottone, P. L., Color reproduction scheme for Kodak Organic Light Emitting Diode (OLED) technology, *AIC Color 01*, Rochester, NY, June 24–29, 2001.
9. Rich, D., Critical parameters in the measurement of the color of nonimpact printing, *J. Electronic Imaging*, 2(3), 1993, 23–236, 1993.
10. Zwinkels, J. C., Colour-measuring instruments and their calibration, *Displays,* 16(4), 163–171, 1996.
11. Rolleston, R., Using Shepard's interpolation to build color transformation tables, in *Proc. IS&T/SID's 2nd Color Imaging Conference*, November 1994, 74–77.
12. R. Balasubramanian, Refinement of printer transformations using weighted regression, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts*, J. Bares, Ed., Vol. 2658, 1996, 334–340.
13. Hung, P-C., Colorimetric calibration in electronic imaging devices using a look-up table model and interpolations, *J. Electronic Imaging*, 2(1), 53–61, 1993.
14. Chang, J. Z., Allebach, J. P., and Bouman, C. A., Sequential linear interpolation of multidimensional functions, *IEEE Trans. on Image Processing*, Vol. IP-6, September 1997, 1231–1245.
15. T. Masters, *Practical Neural Network Recipes in C++,* Academic Press, San Diego, CA, 1993.
16. Farin, G., *Curves and Surfaces for Computer Aided Geometric Design — A Practical Guide*, Academic Press, San Diego, CA, 1988.
17. Press, H., Flannery, B. P., and Vetterling, W. T., *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K., 1988.
18. Luo, M. R., Cui, G. and Rigg, B., The development of the CIEDE2000 colour-difference formula, *Color Res. Appl.*, 25, 340–350, 2001.
19. Engeldrum, P. G., *Psychometric Scaling*, Imcotek Press, Winchester, MA, 2000.
20. Sharma, G. and Trussell, H. J., Figures of merit for color scanners, *IEEE Trans. Image Proc.*, 6(7), 990–1001, 1997.
21. Quan, S. and Ohta, N., Optimization of camera spectral sensitivities, in *Proc. IS&T/SID's 8th Color Imaging Conference*, November 2000, 273–278.
22. Knox, K. T., Integrating cavity effect in scanners, in *Proc. IS&T/OSA Optics & Imaging in the Information Age*, Rochester NY, 1996, 156–158.
23. Kang, H. R., Color scanner calibration, *J. Imaging Sci. Technol.*, 36(2), 162–170, 1992.
24. Sharma, G. and Trussell, H. J., Set theoretic estimation in color scanner characterization, *J. Electronic Imaging*, 5(4), 479–489, 1996.
25. Wandell, B. A., *Foundations of Vision*, Sinauer Associates, Sunderland, MA, 1995.

26. Finlayson, G. D., Hordley, S., and Hubel, P. M., Recovering device sensitivities with quadratic programming, in *Proc. IS&T/SID's 6th Color Imaging Conference*, November 1998, 90–95.
27. Hubel, P. M., Sherman, D., and Farrell, J. E., A comparison of methods of sensor spectral sensitivity estimation, in *Proc. IS&T/SID's 2nd Color Imaging Conference*, November 1994, 45–48.
28. Finlayson, G. D. and Drew, M. S., The maximum ignorance assumption with positivity, in *Proc. IS&T/SID's 4th Color Imaging Conference*, November 1996, 202–205.
29. Berns, R. and Shyu, M. J., Colorimetric characterization of a desktop drum scanner using a spectral model, *J. Electronic Imaging*, 4(4), 360–372, 1995.
30. Sharma, G., Target-less scanner color calibration, *J. Imaging Sci. Technol.*, 44(4), 301–307, 2000.
31. Kang, H. R. and Anderson, P. G., Neural network applications to the color scanner and printer calibrations, *J. Electronic Imaging*, 1(2), 125–135, 1992.
32. ISO 17321 (WD4), *Graphic Technology and Photography — Colour Characterisation of Digital Still Cameras (DSCs) Using Colour Targets and Spectral Illumination*.
33. ISO 14525 (FDIS), *Photography — Electronic Still Picture Cameras — Methods of Measuring Opto-Electronic Conversion Functions (OECFs).*
34. Finlayson, G. D. and Drew, M. S., White point preserving color correction, in *Proc. IS&T/SID's 5th Color Imaging Conference*, November 1997, 258–61.
35. Hubel, P. M. et al., Matrix calculations for digital photography, in *Proc. IS&T/SID's 5th Color Imaging Conference*, November 1997, 105–111.
36. Hong, G., Han, B., and Luo, M. R., Colorimetric characterisation for low-end digital camera, in *Proc. 4th International Conference on Imaging Science and Hardcopy*, 2001, 21–24.
37. Hong, G., Luo, M. R., and Rhodes, P. A., A study of digital camera colorimetric characterisation based on polynomial modelling, *Color Res. Appl.*, 26(1), 76–84, 2001.
38. Finlayson, G. D., Drew, M. S., and Funt, B. V., Spectral sharpening: Sensor transformations for improved colour constancy, *J. Opt. Soc. Am.*, 5, 1553–1563, 1994.
39. Berns, R., S. Motta, R. J., and Gorzynski, M. E., CRT Colorimetry, Part 1: Theory and practice, *Color Res. Appli.*, 18(5), 299–314, 1993.
40. Edgar, A. D. and Kasson, J. M., Display Calibration, U.S. Patent No. 5,298,993, issued March 29, 1994.
41. Engeldrum, P. and Hilliard, W., Interactive Method and System for Color Characterization and Calibration of Display Device, U.S. Patent No. 5,638,117, issued June 10, 1997.
42. Ohara, K. et al., Apparatus for Determining a Black Point on a Display Unit and Method of Performing the Same, U.S. Patent No. 6,084,564, issued July 4, 2000.
43. Balasubramanian, R., Braun, K., Buckley, R., and Rolleston, R., Color documents in the Internet era, *The Industrial Physicist*, 16–20, 2001.
44. Gentile, R. S. and Danciu, I. M., Method to estimate the white point on a display device, U.S. Patent No. 6,023,264, issued February 8, 2000.
45. Marcu, G. et al., Color characterization issues for TFTLCD displays, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII*, Eschbach, R. and Marcu, G., Eds., Vol. 4663, 2002, 187–198.

46. Sharma, G., Comparative evaluation of color characterization and gamut of LCDs versus CRTs, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII*, Eschbach, R. and Marcu, G., Eds., Vol. 4663, 2002, 177–186.
47. Kwak, Y. and MacDonald, L. Accurate prediction of colours on liquid crystal displays, in *Proc. IS&T/SID's 9th Color Imaging Conference*, November 2001, 355–359.
48. H. R. Kang, *Color Technology for Electronic Imaging Devices*, SPIE, Bellingham, WA, 1997.
49. Bala, R., What is the chrominance of gray? *Proc. IS&T and SID's 9th Color Imaging Conference*, November 2001, 102–107.
50. Yule, J. A. C., *Principles of Color Reproduction: Applied to Photomechanical Reproduction, Color Photography, and the Ink, Paper, and Other Related Industries*, John Wiley & Sons, New York, 1967.
51. Allen, E., *Optical Radiation Measurements*, Vol. 2., *Color Formulation and Shading,* Academic Press, San Diego, CA, 1980, Chap. 7.
52. Berns, R. S., Spectral modeling of a dye diffusion thermal transfer printer, *J. Electronic Imaging*, 2(4), 359–370, 1993.
53. Parton, K. H. and Berns, R. S., Color modeling ink-jet ink on paper using Kubelka–Munk theory, in *Proc. 7th Int. Congress on Advances in Non-Impact Printing Technologies*, 1992, 271–280.
54. Kang, H. R., Kubelka–Munk modeling of ink jet ink mixing, *J. Imaging Technol.,* 17, 76–83, 1991.
55. Saunderson, J. L., Calculation of the color of pigmented plastics, *J. Optical Soc. Am.*, 32, 727–736, 1942.
56. Neugebauer, H. E. J., Die Theoretischen Grandlagen des Mehrfarben-edruckes, *Zeitschrift Wissenschaften Photography,* 73–89, 1937.
57. Rolleston, R. and Balasubramanian, R., Accuracy of various types of Neugebauer models, in *Proc. IS&T and SID's 1st Color Imaging Conference: Transforms and Transportability of Color*, November 1993, 32–37.
58. Viggiano, J. A. S., Modeling the color of multi-colored halftones, in *Proc. TAGA,* 44–62, 1990.
59. Balasubramanian, R., Optimization of the spectral Neugebauer model for printer characterization, *J. Electronic Imaging*, 8(2), 156–166, 1999.
60. Yule, J. A. C. and Nielsen, W. J., The penetration of light into paper and its effect on halftone reproduction, in *Proc. TAGA*, 65–76, 1951.
61. Maltz, M., Light scattering in xerographic images, *J. Appl. Photogr. Eng.*, 9(3), 83–89, 1983.
62. Gustavson, S. and Kruse, B., 3D modelling of light diffusion in paper, in *Proc. TAGA*, 2, 848–855, 1995.
63. Heuberger, K. J., Jing, Z. M., and Persiev, S., Color transformations and lookup tables, in *Proc. TAGA/ISCC*, 2, 863–881, 1992.
64. Arney, J. S., Engeldrum, P. G., and Zeng, H., An expanded Murray–Davies model of tone reproduction in halftone imaging, *J. Imaging Sci. Technol.*, 39(6), 502–508, 1995.
65. Lee, B. et al., Estimation of the Neugebauer model of a halftone printer and its application, in *Proc. IS&T/OSA Annu. Conf., Optics & Imaging in the Information Age,* 1997, 610–613.

66. Iino, K. and Berns, R. S., A spectral based model of color printing that compensates for optical interactions of multiple inks, AIC Color 97, in *Proc. 8th Congress International Colour Association*, 1997, 610–613.
67. Hau, C. and Huang, K., Advanced cellular YNSN printer model, in *Proc. IS&T/SID's 5th Color Imaging Conf.,* November 1997, 231–234.
68. Iino, K. and Berns, R. S., Building color management modules using linear optimization, I. Desktop color system, *J. Imaging Sci. Tech*. 42(1), 79–94, 1998.
69. Iino, K. and Berns, R. S., Building color management modules using linear optimization, II. Prepress system for offset printing, *J. Imaging Sci. Technol*., 42(2), 99–114, 1998.
70. Agar, A. U. and Allebach, J. P., An iterative cellular YNSN method for color printer characterization, in *Proc. IS&T/SID's 6th Color Imaging Conference*, November 1998, 197–200.
71. Xia, M. et al., End-to-end color printer calibration by total least squares regression, *IEEE Trans. Image Proc*., 8(5), 700–716, 1999.
72. Balasubramanian, R., Reducing the cost of lookup table based color transformations, *J. Imaging Sci. Technol*., 44(4), 321–327, 2000.
73. Herzog, P., A new approach to printer calibration based on nested gamut shells, in *Proc. IS&T/SID's 5th Color Imaging Conference*, November 1997, 245–249.
74. Pobboravsky, I. and Pearson, M., Computation of dot areas required to match a colorimetrically specified color using the modified Neugebauer equations, in *Proc. TAGA*, 65–77, 1972.
75. Mahy, M. and Delabastita, P., Inversion of the Neugebauer equations, *Color Res. Appl.,* 21(6), 404–411, 1996.
76. Holub, R., Pearson, C., and Kearsley, W., The black printer, *J. Imaging Technol.,* 15(4), 149–158, 1989.
77. Mahy, M., Color Separation Method and Apparatus for Same, U.S. Patent No. 5,878,195, issued March 2, 1999.
78. Balasubramanian, R. and Eschbach, R., Reducing multi-separation color moire via a variable undercolor removal and gray-component replacement strategy, *J. Imaging Sci. Technol*., 45(2), 152–160, 2001.
79. Cholewo, T. J., Printer model inversion by constrained optimization, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V*, R. Eschbach and G. Marcu, Eds., Vol. 3963, San Jose, CA, 2000, 349–357.
80. Kueppers, H., Printing Process where Each Incremental Area is Divided into a Chromatic Area and an Achromatic Area and Wherein the Achromatic Areas Are Printed in Black and White and the Chromatic Areas are Printed in Color Sub-sections, U.S. Patent No. 4,812,899, issued March 14, 1989.
81. Ostromoukhov V., Chromaticity gamut enhancement by heptatone multicolor printing, in *Proc. SPIE, Device Independent Color Imaging and Imaging Systems Integration*, Motta R. J. and Berberian, H. A., Eds., Vol. 1909, 1993, 139–151.
82. Boll, H., A color to colorant transformation for a seven ink process, in *Proc. SPIE, Device Independent Color Imaging*, Walowit E., Ed., Vol. 2170, 1994, 108–118.
83. Balasubramanian, R., System for Printing Color Images with Extra Colorants in Addition to Primary Colorants, U.S. Patent No. 5,870,530, issued Feb. 9, 1999.

84. Cui, C. and Weed, S., Measurement problems for overhead projection transparency printing color calibration, in *Proc. IS&T/SID's 9th Color Imaging Conference*, November 2001, 303–309.
85. Vrhel, M. J., Gershon, R., and Iwan, L. S., Measurement and analysis of object reflectance spectra, *Color Res. Appl.*, 19(1), 4–9, 1991.
86. Keusen, T., Multispectral color system with an encoding format compatible with the conventional tristimulus model, *J. Imaging Sci. Technol.*, 40(6), 510–515, 1996.
87. Tzeng, D. Y. and Berns, R. S., Spectral-based six-color separation minimizing metamerism, in *Proc. IS&T/SID's 8th Color Imaging Conference*, November 2000, 342–347.
88. Tzeng, D. Y. and Berns, R. S., Spectral-based ink selection for multiple-ink printing I. Colorants estimation of original objects, in *Proc. IS&T/SID's 6th Color Imaging Conference*, November 1998, 106–111.
89. Tzeng, D. Y. and Berns, R. S., Spectral-based ink selection for multiple-ink printing II. Optimal ink selection, in *Proc. IS&T/SID's 7th Color Imaging Conference*, November 1999, 182–187.
90. Rosen, M. R. et al., Color management within a spectral image visualization tool, in *Proc. IS&T/SID's 8th Color Imaging Conference*, November 2000, 75–80.
91. Cholewo, T. J., Conversion between CMYK spaces preserving black separation, in *Proc. IS&T/SID's 8th Color Imaging Conference*, November 2000, 257–261.
92. Zeng, H., CMYK transformation with black preservation in color management system, in *Proc. SPIE, Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII*, Eschbach, R. and Marcu, G., Eds., Vol. 4663, 2002, 143–149.
93. Noble, B. and Daniel, J. W., *Applied Linear Algebra*, 2nd ed., Chapter 9, Prentice-Hall, Englewood Cliffs, NJ, 1977, 323–330.

appendix 5.A

Least-squares optimization

Given a $T \times m$ matrix $\mathbf{D}$ of m-dimensional input data points, and a $T \times 1$ vector $\mathbf{c}$ of one-dimensional output data points, we wish to find the optimal $m \times 1$ coefficient vector $\mathbf{a}$ that minimizes the squared error

$$E = \|\mathbf{c} - \mathbf{D}\,\mathbf{a}\|^2 \tag{5.A.1}$$

where $\|\ \|^2$ denotes the L^2 norm or vector length. We can write the error in matrix-vector notation as

$$E = [\mathbf{c} - \mathbf{D}\,\mathbf{a}]^t[\mathbf{c} - \mathbf{D}\,\mathbf{a}] = \mathbf{c}^t\mathbf{c} - 2\mathbf{c}^t\mathbf{D}\mathbf{a} + \mathbf{a}^t\mathbf{D}^t\mathbf{D}\mathbf{a} \tag{5.A.2}$$

The $\mathbf{a}$ that minimizes E is found by differentiating with respect to $\mathbf{a}$ and setting to 0.

$$\frac{\partial E}{\partial \mathbf{a}} = 2\boldsymbol{D}^t\boldsymbol{D}\boldsymbol{a} - 2\boldsymbol{D}^t\boldsymbol{c} = 0 \tag{5.A.3}$$

This leads to

$$\mathbf{a} = (\mathbf{D}^t\mathbf{D})^{-1}\mathbf{D}^t\mathbf{c} \tag{5.A.4}$$

To extend to n-dimensional output, vector $\mathbf{c}$ is replaced by a $T \times n$ matrix $\mathbf{C}$. The foregoing analysis is applied to each column of $\mathbf{C}$, and the resulting linear transformation is now an $\mathbf{m} \times \mathbf{n}$ matrix $\mathbf{A}$ rather than vector $\mathbf{a}$. This results in Equation 5.14.

Direct inversion of the matrix $\mathbf{D}^t\mathbf{D}$ in Equation 5.A.4 can result in numerically unstable solutions, particularly if the system is noisy or ill conditioned. A more robust approach is to use singular value decomposition (SVD). A theorem in linear algebra states that any $T \times m$ matrix $\mathbf{D}$ of rank r can be represented by SVD, given by

$$\mathbf{D} = \mathbf{U}\Sigma\mathbf{V}^t \tag{5.A.5}$$

where $\mathbf{U}$ is a $T \times T$ unitary matrix whose columns $\mathbf{u}_1, \ldots, \mathbf{u}_T$ are the orthonormal eigenvectors of $\mathbf{D}^t\mathbf{D}$; $\mathbf{V}$ is an $m \times m$ unitary matrix whose columns $\mathbf{v}_1, \ldots, \mathbf{v}_m$ are the orthonormal eigenvectors of $\mathbf{D}\mathbf{D}^t$; and Σ is a $T \times m$ matrix given by

$$\sum = \begin{bmatrix} \Delta & 0 \\ 0 & 0 \end{bmatrix} \tag{5.A.6}$$

where Δ is a diagonal $r \times r$ matrix. Proof of this theorem, which is beyond the scope of this chapter, is found in Noble and Daniel.[93]

The diagonal entries, $\sigma_1, \ldots, \sigma_R$, of Δ are the singular values of $\mathbf{D}$. Equation 5.A.5 can be written in series form as

$$\boldsymbol{D} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^t \tag{5.A.7}$$

Substituting Equation 5.A.7 into Equation 5.A.4, we get

$$\mathbf{a} = \sum_{i=1}^{r} \sigma_i^{-1} \mathbf{v}_i \mathbf{u}_i^t \mathbf{c} \tag{5.A.8}$$

An ill-conditioned or noisy system will result in some σ_i taking on very small values, thus resulting in an unstable solution to Equation 5.A.8. Singular value decomposition handles this situation gracefully. A stable least-squares solution can be arrived at by simply eliminating terms corresponding to very small σ_i in the summation of Equation 5.A.8.

A thorough formulation of SVD is given in Reference 93. C software for computing SVD and using it to solve the least-squares problem is provided in Chapter 2 of Reference 17. Popular mathematical packages such as Matlab™ also offer SVD based matrix inversion and least-squares solutions to linear systems.

appendix 5.B

Derivation of 3 × 3 matrix from display RGB to XYZ given white point and chromaticities of the primaries

Given chromaticity coordinates of the three primaries, $[x_R, y_R]$, $[x_G, y_G]$, $[x_B, y_B]$, and the tristimulus values of the white point, $[X_w, Y_w, Z_w]$, the goal is to derive the 3×3 matrix $\mathbf{A}_{CRT}$ that maps display RGB to XYZ as in Equation 5.55.

Assign an arbitrary value $Y_R = Y_G = Y_B = 1$ for the luminance of the three primaries. This provides three-dimensional xyY descriptors for the primaries.

Convert the xyY coordinates of each primary to XYZ space as follows:

$$X'_R = \frac{x_R}{y_R}, \quad Y'_R = 1, \quad Z'_R = \left(\frac{1 - x_R - y_R}{y_R}\right) = \frac{z_R}{y_R} \tag{5.B.1}$$

Analogous expressions apply for the green and blue primaries. This defines a matrix $\mathbf{A}'$ given by

$$\mathbf{A}' = \begin{bmatrix} X'_R & X'_G & X'_B \\ Y'_R & Y'_G & Y'_B \\ Z'_R & Z'_G & Z'_B \end{bmatrix} = \begin{bmatrix} x_R/y_R & x_G/y_G & x_B/y_B \\ 1 & 1 & 1 \\ z_R/y_R & z_G/y_G & z_B/y_B \end{bmatrix} \tag{5.B.2}$$

We now have to scale the column vectors of $\mathbf{A}'$ so that an input of RGB = [1, 1, 1] results in the desired white point $[X_w, Y_w, Z_w]$. This is done by the following operation:

$$\mathbf{A}_{CRT} = \mathbf{A}' * diag(\mathbf{A}'^{-1} * \mathbf{W}) \tag{5.B.3}$$

where **W** is the white vector.