©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.20.COLOR-307
Reprinted from
Journal of Imaging Science and Technology® 60(1): 010401-1–010401-10, 2016.
© Society for Imaging Science and Technology 2016

Developing Calibrating Curves for a Trilinear Interpolation Model During Display Characterization

Bangyong Sun
College of Printing and Packaging, Xi'an University of Technology, Xi'an 710048, China
Norwegian Colour and Visual Computing Laboratory, Norwegian University of Science and Technology, N-2802 Gjøvik, Norway
E-mail: bangyong.sun@hig.no

Jon Y. Hardeberg▲
Norwegian Colour and Visual Computing Laboratory, Norwegian University of Science and Technology, N-2802 Gjøvik, Norway

Congjun Cao
College of Printing and Packaging, Xi'an University of Technology, Xi'an 710048, China

Abstract. *Trilinear interpolation is a method of multivariate interpolation on a three-dimensional regular grid. It approximates the value of an intermediate point using data on the lattice points, and thus is frequently used for display characterization with 3D lookup tables (3D LUTs). However, large color errors are usually caused by the nonlinear relationship between the source RGB space and the destination CIELAB space. In this article the display characterization is improved by modifying the traditional trilinear interpolation model. First, the Yule–Nielsen n-factor is applied to the destination functions, for the purpose of reducing the nonlinearity between the source and destination color spaces. Afterward, different calibrating curves are developed to calculate the effective values of the source RGB values. The input/source RGB values are usually called nominal values, and the effective values can be regarded as the optimized RGB values which improve the matching degree of the predicted and measured destination CIELAB values. In experiment, a Toshiba M5 liquid crystal display is characterized by using the modified trilinear interpolation model, and the forward and inverse characterization errors of different methods are calculated and compared. The evaluation results demonstrate that both the average and the maximum color errors have significantly decreased when calibrating curve III (one of the three types of curves developed) is employed in combination with the optimal n-factor. Thus, the method of developing effective calibrating curves and finding optimal n-factors proposed in this article can be adopted during display characterization.* © *2016 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2016.60.1.010401]

INTRODUCTION

Color characterization defines the relationship between device color space and the measured CIE (Commission internationale de l'éclairage) system color space. For a color display it mainly defines the relationship between the assigned RGB (abbreviation of red, green, blue) signals on disk and the corresponding CIE measurements achieved with colorimeters or spectrophotometers.[1] Two color conversions are involved in the display characterization. The forward process converts the RGB values into CIE values, while the inverse process calculates the CIE colors' separation of RGB values. Because the same RGB signals displayed on different monitors usually yield various CIE values and the relationship between those two spaces is extremely nonlinear, the forward and inverse characterization models are not easily established.[2,3]

In color management systems, there are mainly two types of ICC (International Color Consortium) profiles for modern display devices.[4] One type uses tone reproducing curves to first linearize the input colors and then apply a 3 × 3 matrix to perform the color conversion.[5,6] Another type of display profile employs interpolation models to carry out color transformation within the unit cube of the 3D lookup tables (3D LUTs).[7] The latter is more precise as a large quantity of known colorimetric values can be stored on the LUT lattice points. Actually, in the management workflow, only certain parts of color patches are displayed and measured during display characterization; the majority of the forward and inverse color data are estimated from the measurements. Consequently, it is very important to establish a precise nonlinear model to obtain the relationship between RGB and CIE colors, for the purpose of building accurate ICC profiles and performing color conversion with those ICC profiles.

Many nonlinear mathematical models can be utilized during display characterization, such as polynomial regression, neural network, three-dimensional interpolation methods, etc.[8–10] Within the polynomial regression or neural network models,[11,12] one estimated relationship between RGB and CIE spaces is calculated, and the color conversion process is based on this obtained relationship. Because the color conversion is dependent on only the one relationship, large errors are usually generated, especially for those

▲ IS&T Member.

Received June 30, 2015; accepted for publication Oct. 15, 2015; published online Dec. 03, 2015. Associate Editor: Chunghui Kuo.
1062-3701/2016/60(1)/010401/10/$25.00

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.20.COLOR-307

points near the gamut boundary. The three-dimensional interpolation method[13] is the default color transformation model for most display and output devices within the ICC color management workflow.[14,15] It consists of trilinear, tetrahedral, pyramid, prism interpolations, according to cube slicing algorithms.[8] Compared with other methods, the 3D interpolation is more stable because it always employs several nearest samples for color prediction. Consequently, the display characterization precision is highly dependent on the size of the lookup tables and the accuracy of interpolation. However, the number of color patches and the 3D LUTs cannot be infinitely many considering the measuring time. It is therefore necessary and important to improve the interpolation accuracy with limited sample colors.

In this article, one of the widely used 3D interpolation techniques, trilinear interpolation, is analyzed and improved during display characterization. The second section derives the interpolation error, which is usually caused by the nonlinearity between the source and the destination spaces,[16] and the Yule–Nielsen n-factor[17,18] is introduced to modify the nonlinearity. Nevertheless, the interpolation accuracy does not improve noticeably as the average color error is reduced by merely 0.58ΔE from 3.79ΔE to 3.21ΔE even when the optimal n-factor is used. In order to effectively correct the trilinear interpolation model's nonlinearity, in the third section we propose a method to develop calibration curves for the red, green and blue single channels, which convert the nominal RGB values into their effective values. Three types of calibration curves are developed and evaluated. Curve I is obtained by optimizing the center points of red, green and blue channels independently; curve II is generated by using the center points on the diagonal line of the RGB cube, and the center points' effective values are calculated by comparing the predicted and measured CIE values. Curve III's sample points are similar to those of curve II, but the center points' effective values are computed by minimizing the color error of many uniformly distributed testing samples. In the fourth section we describe the experiment where the forward and inverse characterizations based on trilinear interpolation are evaluated. Within the experiment results, the characterization accuracy improves significantly when curve III and the optimal n-factor are employed, which indicates that the method of improving linearity by calibrating curves and n-factors can be applied during color characterization. Finally, conclusions are drawn in the fifth section.

TRILINEAR INTERPOLATION MODEL FOR DISPLAY CHARACTERIZATION

Three-dimensional interpolation models are usually applied to perform color conversion within forward or inverse 3D LUTs. Several models are involved in the 3D interpolation method, such as trilinear, tetrahedron, prism, pyramid, etc. In fact, the calculation procedures of those models are very similar, even the accuracy of them is also approximate. In this article, trilinear interpolation is selected as the characterization model, which employs all eight lattice points of the unit cube. In this section, we analyze the trilinear interpolation errors caused by the nonlinearities, and find the solutions for improving interpolation accuracy.

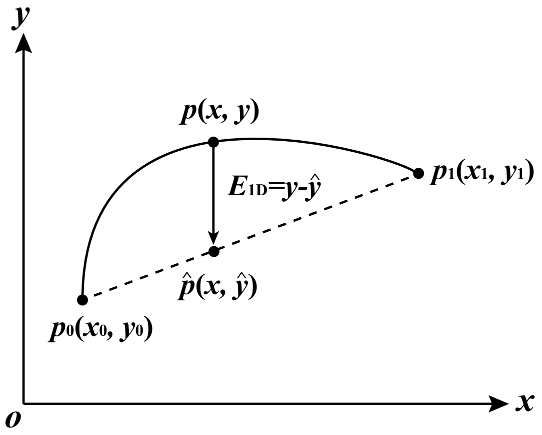


Figure 1. The error in 1D linear interpolation.

Trilinear Interpolation Error Analysis

Trilinear interpolation is usually performed on the 3D LUTs. Within the forward display characterization LUTs, RGB is the source space, and CIELAB is often the destination space which is stored on the RGB vertexes. For any inner point $p(R_p, G_p, B_p)$ that is enclosed by a unit cube with eight vertexes $K(R_0, G_0, B_0)$, $R(R_1, G_0, B_0)$, $G(R_0, G_1, B_0)$, $B(R_0, G_0, B_1)$, $C(R_0, G_1, B_1)$, $M(R_1, G_0, B_1)$, $Y(R_1, G_1, B_0)$, $W(R_1, G_1, B_1)$, the coordinates must satisfy the inequalities $R_0 \leq R_p \leq R_1$, $G_0 \leq G_p \leq G_1$, and $B_0 \leq B_p \leq B_1$. Then, p's CIELAB values can be predicted by using trilinear interpolation as follows:

$$f(R_p, G_p, B_p) = c_0 + c_1\Delta R + c_2\Delta G + c_3\Delta B + c_4\Delta R\Delta G + c_5\Delta G\Delta B + c_6\Delta B\Delta R + c_7\Delta R\Delta G\Delta B, \quad (1)$$

where ΔR, ΔG, and ΔB are the relative distances of point p from the origin point $K(R_0, G_0, B_0)$ in the R, G, and B directions, while f is a scalar-valued function in the output space, and the coefficients c_i are determined by the values of the unit cube vertexes listed in Table I.

$$\Delta R = \frac{R_p - R_0}{h} = \frac{R_p - R_0}{R_1 - R_0},$$
$$\Delta G = \frac{G_p - G_0}{h} = \frac{G_p - G_0}{G_1 - G_0}, \quad (2)$$
$$\Delta B = \frac{B_p - B_0}{h} = \frac{B_p - B_0}{B_1 - B_0}.$$

In fact, trilinear interpolation is a multiple application of linear interpolation; in other words, it is derived from 1D linear interpolation and 2D bilinear interpolation which is also the iteration of 1D linear interpolation.[19] Thus, a 1D linear interpolation error will exist in all three types of interpolations. As shown in Figure 1, if point $p(x, y)$ is between two known points $p_0(x_0, y_0)$ and $p_1(x_1, y_1)$, the approximated value $\hat{y}$ can be calculated using 1D linear

Table I. Coefficients of trilinear interpolation.

Coefficients	Definition
c_0	$f(R_K, G_K, B_K)$
c_1	$f(R_R, G_R, B_R) - f(R_K, G_K, B_K)$
c_2	$f(R_G, G_G, B_G) - f(R_K, G_K, B_K)$
c_3	$f(R_B, G_B, B_B) - f(R_K, G_K, B_K)$
c_4	$f(R_Y, G_Y, B_Y) - f(R_G, G_G, B_G)$ $- f(R_R, G_R, B_R) + f(R_K, G_K, B_K)$
c_5	$f(R_C, G_C, B_C) - f(R_B, G_B, B_B)$ $- f(R_G, G_G, B_G) + f(R_K, G_K, B_K)$
c_6	$f(R_M, G_M, B_M) - f(R_B, G_B, B_B)$ $- f(R_R, G_R, B_R) + f(R_K, G_K, B_K)$
c_7	$f(R_W, G_W, B_W) - f(R_C, G_C, B_C)$ $- f(R_M, G_M, B_M) - f(R_Y, G_Y, B_Y)$ $+ f(R_R, G_R, B_R) + f(R_B, G_B, B_B)$ $+ f(R_G, G_G, B_G) - f(R_K, G_K, B_K)$

interpolation as follows:

$$\hat{y} = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0). \tag{3}$$

Since the relationship among these three points is usually nonlinear, there is an obvious interpolation error between the actual value y and the interpolated value $\hat{y}$:

$$E_{1D}(p, p_0, p_1) = y - \hat{y}. \tag{4}$$

Thereby, when trilinear interpolation is employed to approximate color values within 3D LUTs, the interpolation error will be iterated. As shown in Figure 2, point p is enclosed in a unit cube of 3D LUTs, and it can be interpolated by p_0 and p_1, while p_1 is calculated by p_{01} and p_{11}, and p_{01} is computed from p_{001} and p_{101}. Consequently, the 1D interpolation error exists in every iteration process, and the trilinear interpolation error can be analytically treated as the sum of all those 1D interpolation errors involved, as follows:

$$\begin{aligned} E_{3D}(p, p_{000\sim111}) \equiv\ & E_{1D}(p, p_0, p_1) + E_{1D}(p_0, p_{00}, p_{10}) \\ & + E_{1D}(p_1, p_{01}, p_{11}) + E_{1D}(p_{01}, p_{001}, p_{101}) \\ & + E_{1D}(p_{11}, p_{011}, p_{111}) + E_{1D}(p_{00}, p_{000}, p_{100}) \\ & + E_{1D}(p_{10}, p_{010}, p_{110}), \end{aligned} \tag{5}$$

where E_{3D} represents the expected value of the 3D trilinear error and E_{1D} denotes the 1D interpolation error as in Eq. (4). It is important to note that the actual value of E_{3D} is not the sum of all those E_{1D} errors, Eq. (5) just expresses that the E_{3D} error is highly correlated to all those E_{1D} errors.

Yule–Nielsen Model for Modifying the Trilinear Interpolation's Nonlinearities

As a result of the iterated interpolation error in Eq. (1), the high-order items $\Delta R \Delta G$, $\Delta G \Delta B$, $\Delta R \Delta B$, and $\Delta R \Delta G \Delta B$ result in nonlinearities between the source RGB values and the destination CIELAB values. In order to reduce the influence of nonlinearity, we reformed the trilinear interpolation and corrected it with an exponential function like the Yule–Nielsen model.

Table II. Weighted parameters of trilinear interpolation.

$f(R_{pi}, G_{pi}, B_{pi})$	w_i
$f(R_K, G_K, B_K)$	$(1 - \Delta R)(1 - \Delta G)(1 - \Delta B)$
$f(R_R, G_R, B_R)$	$\Delta R(1 - \Delta G)(1 - \Delta B)$
$f(R_G, G_G, B_G)$	$\Delta G(1 - \Delta R)(1 - \Delta B)$
$f(R_B, G_B, B_B)$	$\Delta B(1 - \Delta R)(1 - \Delta G)$
$f(R_Y, G_Y, B_Y)$	$\Delta R \Delta G(1 - \Delta B)$
$f(R_C, G_C, B_C)$	$\Delta G \Delta B(1 - \Delta R)$
$f(R_M, G_M, B_M)$	$\Delta R \Delta B(1 - \Delta G)$
$f(R_W, G_W, B_W)$	$\Delta R \Delta G \Delta B$

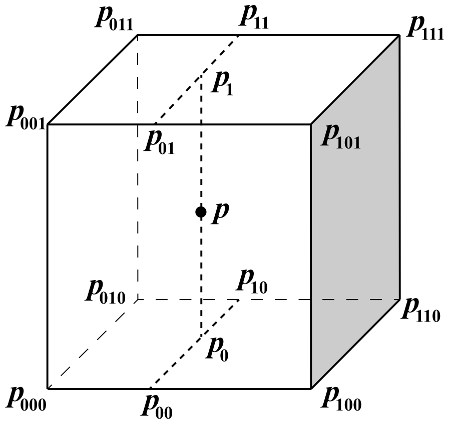


Figure 2. The error in trilinear interpolation.

In Eq. (1) the interpolated function $f(R, G, B)$ is equal to $f(R_0 + \Delta R * h, G_0 + \Delta G * h, B_0 + \Delta B * h)$ with the origin point (R_0, G_0, B_0) in the unit cube, and it is expressed as a group of polynomial terms composed of the ΔR, ΔG, and ΔB. In fact, if those eight cube vertexes are individually considered, Eq. (1) can be reformed to another expression,[16] and the interpolated values are expressed as the weighted sum of the cube vertexes' CIELAB values as follows:[17,20]

$$f(R, G, B) = \sum_{i=1}^{8} w_i f(R_{pi}, G_{pi}, B_{pi}). \tag{6}$$

Equation (6) calculates the same result as Eq. (1), where p_i represents the unit cube's eight vertexes, and w_i means the weighting coefficients of the vertexes which are defined by the offsets ΔR, ΔG, and ΔB listed in Table II.

Equation (6) is similar to the conventional Neugebauer model which predicts the spectral reflectance for printed inks, and the weighting coefficients are all derived from the Demichel equations.[21] In fact, the Neugebauer model is also nonlinear because of the dot gain effect which is usually explained by the ink spreading and lateral light propagation

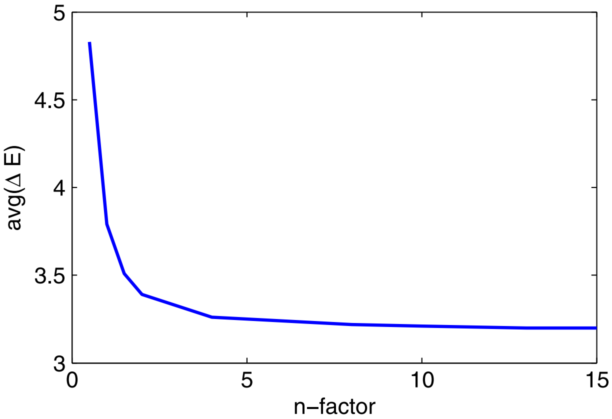


Figure 3. Color difference of trilinear interpolation with different *n*-factors.

in the substrate.[22–24] Viggiano[25] used an n-factor (or so-called n-value) to compensate the dot gain effect, which noticeably improved the linearity of the Neugebauer model. Thus, we try to introduce that exponential function into display characterization, and test whether the linearity and the interpolation accuracy can also be highly improved:

$$\begin{aligned} f(R,G,B) = \Big[& (1-\Delta R)(1-\Delta G)(1-\Delta B) f(K)^{\frac{1}{n}} \\ & + \Delta R(1-\Delta G)(1-\Delta B) f(R)^{\frac{1}{n}} \\ & + (1-\Delta R)\Delta G(1-\Delta B) f(G)^{\frac{1}{n}} \\ & + (1-\Delta R)(1-\Delta G)\Delta B f(B)^{\frac{1}{n}} \\ & + (1-\Delta R)\Delta G \Delta B f(C)^{\frac{1}{n}} \\ & + \Delta R(1-\Delta G)\Delta B f(\lambda M)^{\frac{1}{n}} \\ & + \Delta R \Delta G(1-\Delta B) f(Y)^{\frac{1}{n}} \\ & + \Delta R \Delta G \Delta B f(W)^{\frac{1}{n}} \Big]^{n}. \qquad (7) \end{aligned}$$

Because the n-factor is not a fixed value, the optimal n-value with the minimum error should be determined during color prediction. We search for the optimal n-factor with an iterative algorithm. First, an initial value n_0 is given and the color errors of several testing samples are calculated. The n-factor constantly increases with the new value $(n_0 + \Delta n)$, and the new color errors are calculated simultaneously, where Δn is the presetting offset. The iterative algorithm will stop when the optimal n-value corresponding to the minimum error is found. In this article, we evaluate the performance of different n-factors from 0.5 to 15, and the average color errors of 64 testing samples are shown in Figure 3. This figure demonstrates that the color error decreases rapidly from the initial point $n = 0.5$, and gradually stabilizes at the point $n = 8$, so it is not advisable to use an n-factor below 1 during display characterization. For those n-values greater than 1, one value can be chosen as the optimal n-value where its color error is very close to the minimum and stays almost unchanged from that point. Therefore, we apply $n = 8$ to the display characterization when the color prediction model of Eq. (7) is employed for color conversion.

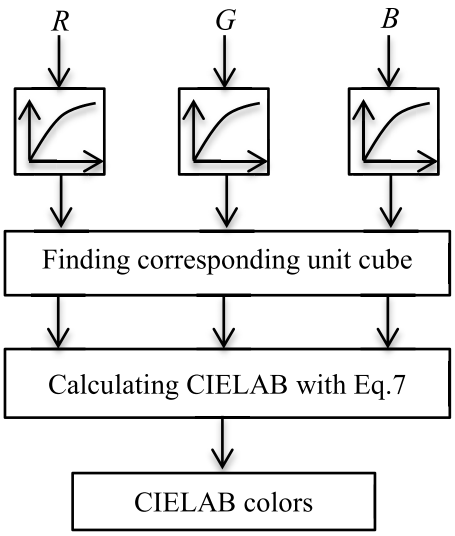


Figure 4. Color predicting workflow with RGB calibration curves.

DEVELOPING THE CALIBRATION CURVES FOR DISPLAY CHARACTERIZATION

The Yule–Nielsen n-factor usually improves the prediction precision greatly for the spectral Murray–Davies model or the Neugebauer model. However, the advantage does not hold for display devices. A major reason may be that the monitor display process is based on additive coloring, not the subtractive coloring of printers, and the dot gain is rather different from the halftone printing process. Consequently, other solutions should be developed to reduce nonlinearity in trilinear interpolation. In this section, calibration curves are proposed to calculate the effective values from the nominal RGB values, in order to correct the nonlinearity with piecewise functions. The forward display characterization workflow using calibrating curves is illustrated in Figure 4.

In Fig. 4, three independent calibrating curves are developed for the red, green and blue channels. The input values R, G, B will turn into the effective values with calibration curves, and then go through the 3D LUTs to find the corresponding unit cube. The color prediction is performed in that unit cube with eight vertexes' information, and the effective values R′, G′, B′ will improve the linearity of the trilinear interpolation model and decrease the characterization error.

It should be noted that the calibration curves are not only used for the forward characterization process, but also for the inverse process. In this article we employ an optimization method to transform the inverse characterization into the iteration process of forward characterization. The practical inverse characterization workflow is shown in Figure 5. A given CIELAB color is first mapped into the gamut of the display, and then converted into the RGB values with the optimization method. As the gamut mapping algorithms are not of great concern in this article, we select the CIELAB colors within the display gamut during inverse characterization. The given CIELAB color is constantly

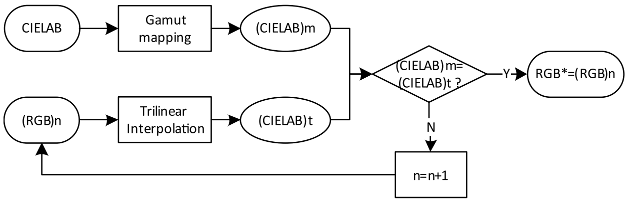


Figure 5. The inverse characterization based on the optimization process.

compared with the trilinear interpolated CIELAB values, and the color difference is calculated simultaneously. When both CIELAB colors match well with a very small color difference generated, the input RGB value can be selected as the input CIELAB color's optimal solution. Thus, the calibration curves determine the display characterization accuracy in both the forward and the inverse directions.

Building the Calibrating Curve for each Channel

In order to modify the nonlinearity of the trilinear interpolation and obtain minimal errors during color prediction, the effective values are usually substituted for the nominal values. The effective values can be defined as the optimal source combinations that create the best matching relationship with the destination values. In this article the calibrating curves are built by using several scattered points with the nominal and calculated effective values; thus, the calculation of the effective values is essential.

Three types of calibrating curves, curve I, curve II, and curve III, are developed with different scattered points and effective value calculation principles. Curve I is built by using several lattice points of the 3D LUTs and some center points between them, and these points are all located in the red, green or blue axes of the RGB cube. For the lattice vertexes, the effective values are equal to the nominal values in order to maintain the calibrated RGB values in the original unit cubes. The center points' effective values for curve I are calculated separately for the red, green, and blue channels, and each point's effective values correspond to the minimum interpolation error. Curve II uses the lattice points and the center point on the diagonal line of the RGB cube, so it only uses one third of the sample points in curve I. The calculation of the center point's effective values is similar to curve I, which just considers the color error of the center point itself. Curve III's scattered points are the same as curve II, but the effective values are calculated by finding the minimal color error of testing samples. The process of building the calibration curves can be described in the following steps.

(1) Create 3D LUT lattice vertexes by dividing R, G, B colors into k parts respectively, and there are $(k+1)^3$ points in the sample dataset S_1:

$$S_1 = \left\{ [r, g, b] \in R \times G \times B \,\middle|\, r \in \left\{0, \frac{1}{k}, \frac{2}{k}, \dots, 1\right\}, \right.$$
$$\left. g \in \left\{0, \frac{1}{k}, \frac{2}{k}, \dots, 1\right\}, b \in \left\{0, \frac{1}{k}, \frac{2}{k}, \dots, 1\right\} \right\}. \tag{8}$$

(2) Generate center-point dataset S_2 as follows, and those $k \times 3$ points are used for establishing calibrating curves I:

$$S_2 = \left\{ [r, g, b] \in R \times G \times B \,\middle|\, \begin{array}{l} r \in \left\{\frac{1}{2k}, \frac{3}{2k}, \dots, \frac{2k-1}{2k}\right\}, g = b = 0 \\ g \in \left\{\frac{1}{2k}, \frac{3}{2k}, \dots, \frac{2k-1}{2k}\right\}, r = b = 0 \\ b \in \left\{\frac{1}{2k}, \frac{3}{2k}, \dots, \frac{2k-1}{2k}\right\}, r = g = 0 \end{array} \right\}. \tag{9}$$

(3) Build center-point dataset S_3 as follows, which is used for establishing calibrating curves II and III:

$$S_3 = \left\{ [r, g, b] \in R \times G \times B \,\middle|\, r \in \left\{\frac{1}{2k}, \frac{3}{2k}, \dots, \frac{2k-1}{2k}\right\}, \right.$$
$$\left. r = g = b \right\}. \tag{10}$$

(4) Generate a quantity of testing samples S_4 for display characterization model evaluation and Curve III's effective value calculation, where the RGB values are randomly distributed in the RGB cube.

(5) With the RGB values of the sample dataset, a quantity of colors are displayed on the monitor and the CIELAB values are measured, then the 3D lookup tables are established.

(6) Calculate the effective values of those center points in S_2 and S_3, and then build the calibration curves I, II and III.

The calibration curves consist of the LUT lattice points and center points, and for every channel the nominal values are arranged in sequence of $0, 1/2k, 2/2k, 3/2k, \dots, 1$. Because the effective values of the lattice points are equal to the nominal values, the calibration curves for each channel can be described as a piecewise function. Within each part of this function, three points are involved with the nominal values $i/2k$, $(i+1)/2k$, $(i+2)/2k$, $0 \le i \le 2k-2$, and their effective values are $i/2k$, $((i+1)/2k)'$, $(i+2)/2k$, where $((i+1)/2k)'$ represents the effective value of center point $(i+1)/2k$. For one segment of this piecewise function with these three known points, quadratic Lagrange interpolation can be employed to calculate any point's effective values from the given nominal value. Quadratic Lagrange interpolation with three given points (x_0, y_0), (x_1, y_1), (x_2, y_2) is expressed as follows:

$$y = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \tag{11}$$

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.20.COLOR-307

Calculation of the Center Points' Effective Values

Based on the trilinear interpolation, the constrained optimization method is applied to calculate the center points' effective values, and there are three main steps involved, which are selecting a continuous objective function, defining the feasible region, and calculating the optimal solutions. We use the modified trilinear interpolation of Eq. (7) as the forward characterization model. Since the center points' nominal values and the measured CIELAB values are given, the effective values can be found by minimizing the preferred objective function such as one form of color difference:

$$RGB^* = \arg\min \Delta E(f(RGB, n), Lab_m), \qquad (12)$$

where RGB^* represents the optimized effective values, n is the Yule–Nielsen n-factor, Lab_m represents the measured CIELAB colors, and ΔE is the color error between the interpolated and measured CIELAB values. Actually, there are many forms of color difference formulas,[26] such as ΔE_{76}, ΔE_{94}, ΔE_{00}, and so on, while the simplest color difference equation of CIE76 is expressed as follows:

$$\Delta E^*_{ab} = \sqrt{(L^* - \hat{L}^*)^2 + (a^* - \hat{a}^*)^2 + (b^* - \hat{b}^*)^2}, \qquad (13)$$

where (L^*, a^*, b^*) and $(\hat{L}^*, \hat{a}^*, \hat{b}^*)$ are the measured and the interpolated CIELAB values, respectively.

In fact, it is not robust to just use the color error objective function. If the effective values differ greatly from the nominal values, the calibrating curves are usually not stable and sometimes even perform worse than the case of no curves. Therefore, from the overall point of view, the Euclidean distance in RGB space between the nominal and effective points is considered:

$$\Delta RGB = \sqrt{(R - R')^2 + (G - G')^2 + (B - B')^2}. \qquad (14)$$

In the above equation, (R, G, B) and (R', G', B') are the nominal and effective values, respectively. Hence, in this article the objective function f_{obj} is a combination of the color difference and the RGB distance:

$$f_{\text{obj}} = \alpha \cdot \Delta E + (1 - \alpha) \cdot \Delta RGB, \qquad (15)$$

where $0 \le \alpha \le 1$, and its value can be defined practically. Since the major purpose of applying the RGB error is to improve the robustness of the objective function, the weight of the color difference of Eq. (13) should be much larger than the RGB error of Eq. (14). In this article the factor α is set as 0.9, which corresponds to the minimum color error. When different α values are applied, the characterization error fluctuates somewhat when α is set between 1 and 0.8, and then becomes much greater in the case of $\alpha < 0.8$. Therefore, the proposed value for α is between 1 and 0.8, but that value range may change somewhat for different displays.

The next step is to define the feasible region of the effective values. Generally, the RGB effective values have definite physical gamut constraints, and they are usually represented as $0 \le R, G, B \le 255$, or normalized as $0 \le R, G, B \le 1$. In order to prevent the effective points from going outside the original unit cubes, further constraints should be imposed. For example, one center point's nominal value is $(R, 0, 0)$, and it is between the unit cube's two lattice vertexes $(R_{\text{low}}, 0, 0)$ and $(R_{\text{up}}, 0, 0)$, then the feasible region should be further constrained as

$$R_{\text{low}} \le R' \le R_{\text{up}};\ G' = 0;\ B' = 0. \qquad (16)$$

Table III. Effective values of the center points for curve I.

Curve I					
Nominal *R*	Effective *R′*	Nominal *G*	Effective *G′*	Nominal *B*	Effective *B′*
(32, 0, 0)	(22, 0, 0)	(0, 32, 0)	(0, 22, 0)	(0, 0, 32)	(0, 0, 27)
(96, 0, 0)	(101, 0, 0)	(0, 96, 0)	(0, 96, 0)	(0, 0, 96)	(0, 0, 101)
(160, 0, 0)	(157, 0, 0)	(0, 160, 0)	(0, 158, 0)	(0, 0, 160)	(0, 0, 159)
(224, 0, 0)	(217, 0, 0)	(0, 224, 0)	(0, 221, 0)	(0, 0, 224)	(0, 0, 216)

Finally, many algorithms can be used to solve constrained optimization problems, such as the steepest descent method, the simplified Newton method, the conjugate gradient method, and the variable metric method; we use the bound-constrained optimization algorithm *L-BFGS-B* as no other complex constraints are involved.[27]

EXPERIMENTS AND ANALYSIS

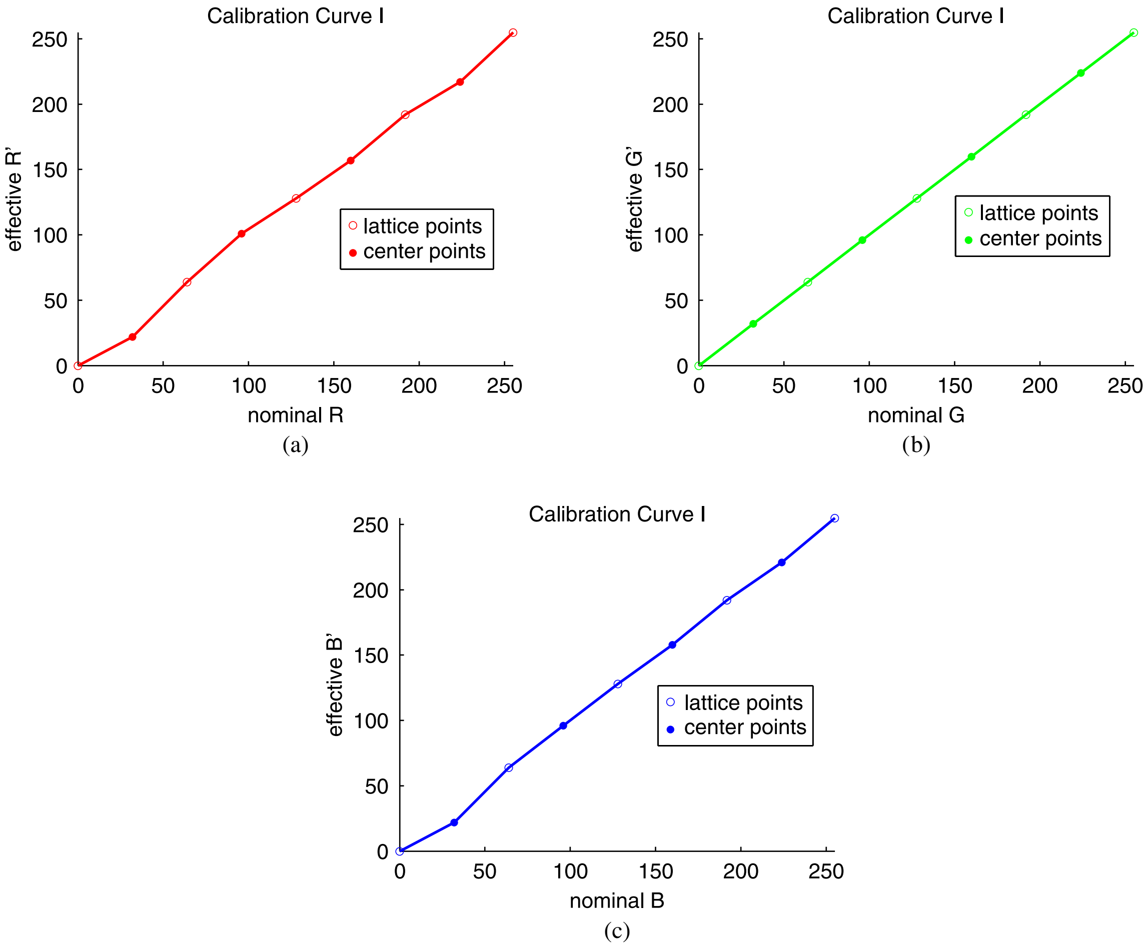
Calculation of the Three Calibration Curves

In the experiments, the modified trilinear interpolation based on the calibrating curves and the Yule–Nielsen n-factor is employed to characterize a Toshiba M5 liquid crystal display in the forward and inverse directions, and the white point of the display is set as CIE D65. According to Eqs. (8) and (10), we set $k = 4$ to generate the training samples, and use an *X-Rite* DTP94 spectrophotometer to measure the CIELAB values of the displayed patches. Every sample is measured three times, and the average of the three measured CIELAB values is utilized.

Because the red, green, and blue channels are divided into four parts, there are 64 unit cubes and $5^3 = 125$ vertexes within the RGB 3D LUTs. According to the definition of the center points for curves I, II and III, the lattice coordinates for each channel are in the set $\{0, 64, 128, 192, 255\}$, and there are 12 center points in curve I with the nominal RGB values (32, 0, 0), (96, 0, 0), (160, 0, 0), (224, 0, 0), (0, 32, 0), (0, 96, 0), (0, 160, 0), (0, 224, 0), (0, 0, 32), (0, 0, 96), and (0, 0, 224), while for curves II and III, the center points' nominal RGB values are (32, 32, 32), (96, 96, 96), (160, 160, 160), and (224, 224, 224).

During the calculation of the effective values for curve I, one of the RGB channel's feasible regions is constrained within the unit cube, and the other two are set as 0. By interpolating and searching for the minimal objective function value, the center effective values for the red, green, and blue channels are obtained, and are listed in Table III, and the curves are depicted in Figure 6 combined with the lattice points.

DOI: 10.2352/ISSN.2470-1173.2016.20.COLOR-307



(a)

(b)

(c)

Figure 6. Calibration curve I for the red, green, and blue channels.

Table IV. Effective values of the center points for curves II and III.

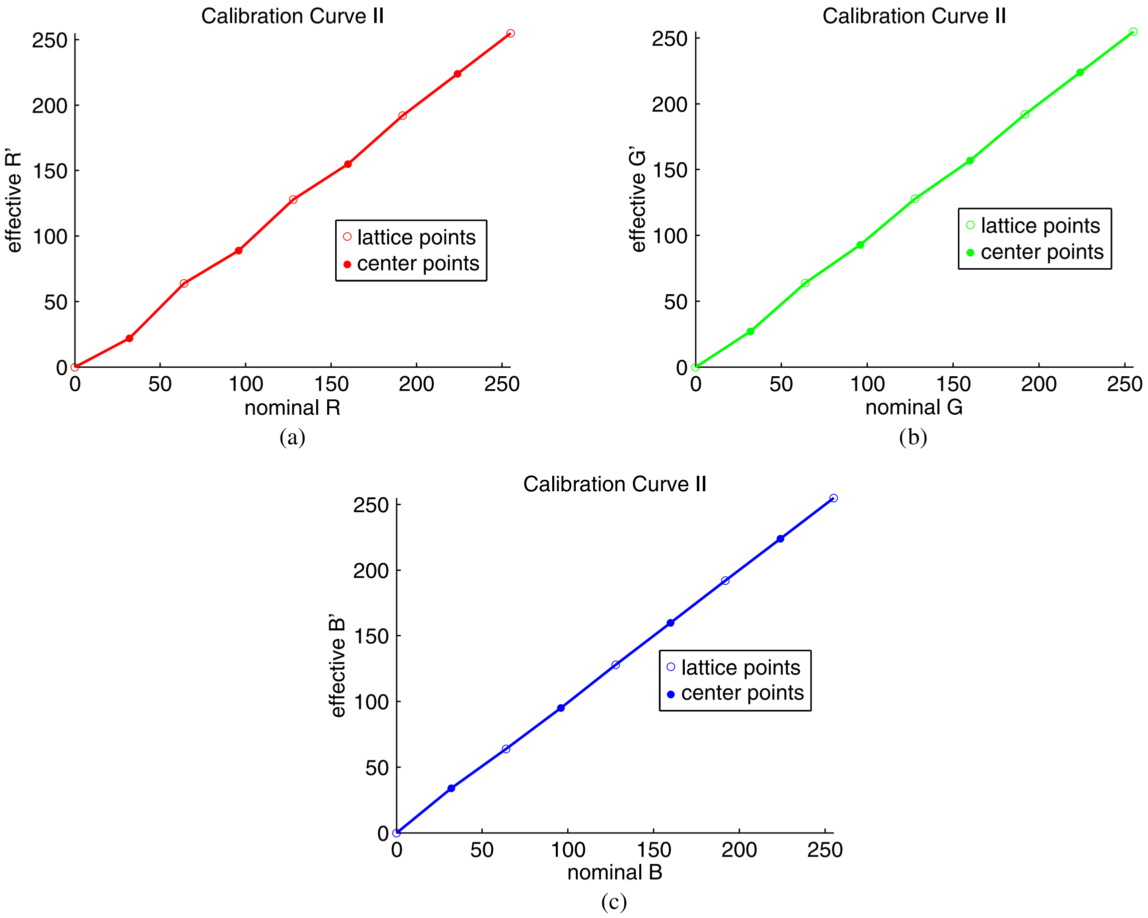
Curve II		Curve III	
Nominal points	Effective points	Nominal points	Effective points
(32, 32, 32)	(22, 27, 34)	(32, 32, 32)	(22, 22, 26)
(96, 96, 96)	(89, 93, 95)	(96, 96, 96)	(91, 92, 95)
(160, 160, 160)	(155, 157, 160)	(160, 160, 160)	(155, 156, 159)
(224, 224, 224)	(224, 224, 224)	(224, 224, 224)	(220, 225, 228)

For curves II and III, only four center points on a diagonal line are involved. The effective values for curve II are calculated by finding the minimum error of the center point itself, while for curve III the average error of all of the testing points is considered, and there are 64 randomly distributed testing points for calculating curve III's effective values. The effective values for curves II and III are shown in Table IV, and the calibrating curves for the individual channels are depicted in Figures 7 and 8.

Forward and Inverse Characterization Error Analysis

The above three calibrating curves are evaluated with 64 testing samples in the form of color differences. The testing RGB values are in the set {20, 80, 180, 240}; these values are different from the lattice points and center points. In the forward characterization process, the trilinear interpolation with different curves is first employed to predict the CIELAB colors, and then color differences are calculated with the predicted and measured CIELAB values. The color errors of the trilinear interpolation model with different curves or n-values are listed in Table V. Since many display profiles are based on the 3 × 3 matrix with tone response curves,[28] the matrix method is also employed for characterization accuracy comparison.

In Table V, "no curve" means no calibrating curves are used, so it is the traditional trilinear interpolation with n-factors, while n^* represents the optimal n-value along with its corresponding calibration curve. For the different characterization methods, besides the average and maximum color difference, the standard deviation of the error is also calculated for the purpose of analyzing error distributions. Among all these color conversion methods, the characterization error of the matrix transform method is greater than those of the 3D LUT methods, mainly because the tested display's color performance is not quite linear and the single-channel samples are few. For all of the 125 LUT samples in this experiment, only five colors of each



(a)

(b)

(c)

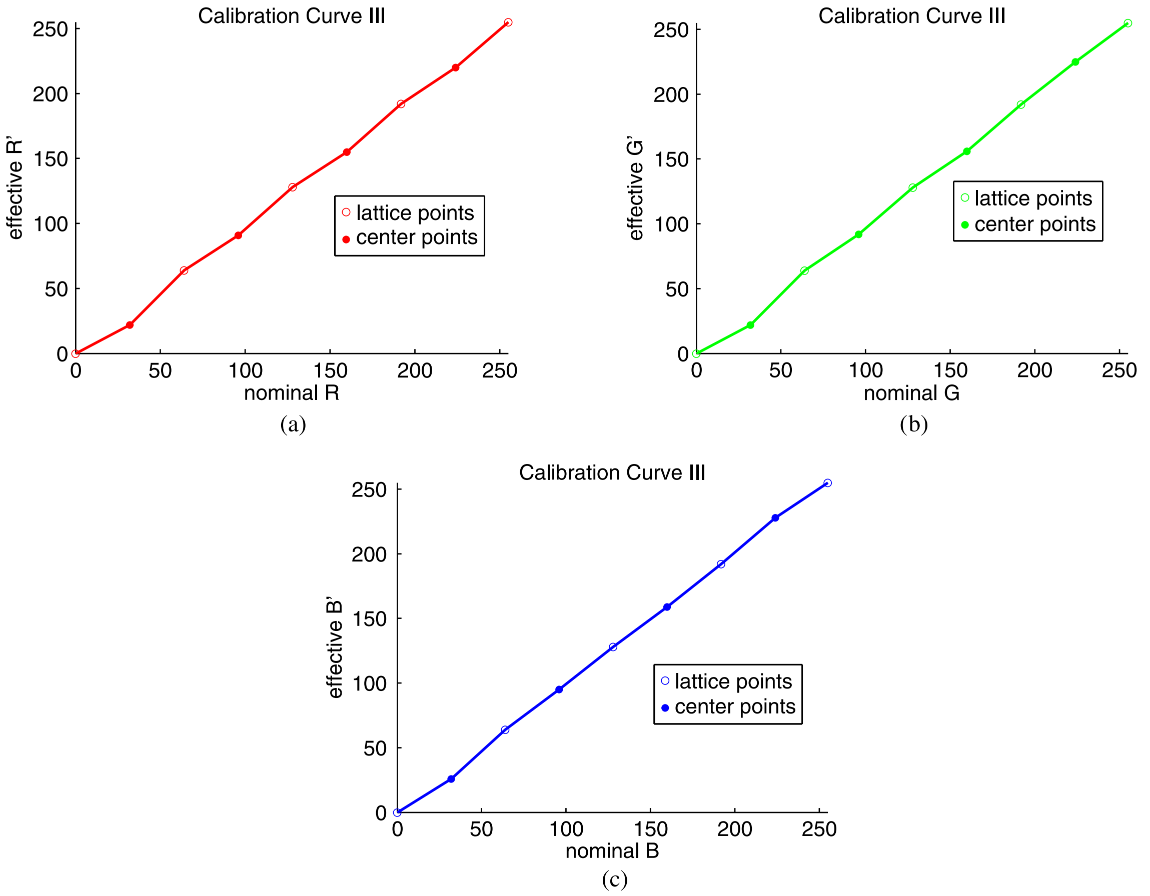
Figure 7. Calibration curve II for the red, green, and blue channels.

Table V. Color differences for various forward characterization algorithms.

Forward characterization	Mean(ΔE)	Max(ΔE)	Std(ΔE)
3 × 3 matrix	5.57	11.31	2.43
No curve ($n = 1$)	3.79	8.02	1.90
No curve ($n^* = 8$)	3.21	7.91	1.99
Curve I ($n = 1$)	3.76	7.21	1.84
Curve I ($n^* = 1.5$)	3.70	7.02	1.84
Curve II ($n = 1$)	2.94	6.58	1.79
Curve II ($n^* = 1.5$)	2.88	6.56	1.81
Curve III ($n = 1$)	2.30	5.67	1.80
Curve III ($n^* = 1.6$)	2.15	5.65	1.81

single channel can be used to calculate the 1D LUT or gamma value. This fact significantly influences the matrix color conversion process. Besides, the black level subtraction and the channel or spatial independence analysis are also important for matrix transformation characterization, but they are not included in our experiment considering the single-channel samples; just the simple GOG model[28] and the $3 * 3$ matrix are used for color conversion. Actually, many excellent works on matrix transform display characterization have been carried out by Ellen A. Day, Lawrence Taplin, and Roy S. Berns, especially Ref. 29.

While for those LUT methods, the traditional trilinear interpolation produces the largest color errors without applying calibrating curves and the Yule–Nielsen correction, the optimal n-factor reduces the characterization average error for all of the calibration curves compared with the cases of $n = 1$, but the standard deviation of the error does not change much. When the specific calibrating curves in combination with the optimal n-values are applied, curve III performs better than the others. Besides, the color errors for curves II and III are both smaller than those for curve I, which indicates that the center points on the diagonal line should be selected compared with the center points on individual channels. Although these three types of curves consume almost the same time during color conversion, the time for calculating their center points' effective values are significantly different. Curves I, II, and III consume 13 s, 5 s, and 25 s, respectively. Curves I and II apply the same principle for calculating the center points, so curve II is faster since fewer center points are involved. Each effective value of a curve III center point is calculated by comparing all of the 64 testing samples' average error, and that process consumes much more time as massive calculations are involved.



(a)

(b)

(c)

Figure 8. Calibration curve III for the red, green, and blue channels.

According to the workflow in Fig. 5, the inverse characterization based on the trilinear interpolation model is also evaluated. For any given CIELAB color, the trilinear interpolation and optimization method is employed to search for the RGB value that predicts the closest CIELAB color. The RGB color errors for different models during inverse characterization are listed below.

The inverse characterization error is defined in Eq. (12), where all of the RGB values are normalized between 0 and 1. As shown in Table VI, the accuracy comparison for the models with different calibrating curves and n-values is similar to Table V, and the minimum color error is still generated from curve III. Therefore, when curve III in conjunction with the optimal n-factor is applied in the experiment, the nonlinearity of the trilinear interpolation model is highly modified and the best prediction accuracy is obtained.

Table VI. Color differences for various inverse characterization algorithms.

Inverse characterization	Mean(ΔRGB)	Max(ΔRGB)	Std(ΔRGB)
3 × 3 matrix	0.055	0.082	0.0181
No curve ($n = 1$)	0.042	0.067	0.0134
No curve ($n^* = 8$)	0.037	0.068	0.0162
Curve I ($n = 1$)	0.039	0.068	0.0094
Curve I ($n^* = 1.5$)	0.036	0.067	0.0094
Curve II ($n = 1$)	0.035	0.062	0.0094
Curve II ($n^* = 1.5$)	0.033	0.061	0.0094
Curve III ($n = 1$)	0.027	0.058	0.0095
Curve III ($n^* = 1.6$)	0.026	0.056	0.0095

CONCLUSIONS

The trilinear interpolation model is widely used in color characterization with 3D LUT ICC profiles, such as displays or many output devices. In this article, the trilinear interpolation model with different calibrating curves and the Yule–Nielsen n-factor is evaluated. First, the application of trilinear interpolation based on 3D LUTs is introduced, and the problems within the trilinear technique are analyzed, especially the nonlinearities. In order to improve characterization accuracy, the traditional trilinear interpolation is modified by using the Yule–Nielsen n-factor, but the performance is not obviously improved. Thus, calibrating curves are developed to change the RGB nominal values into effective values, which greatly improve the trilinear interpolation model's linearity. In the experiments, the modified trilinear interpolation model with calibration curves and optimal n-factors is evaluated during forward and inverse characterization processes. The experimental

DOI: 10.2352/ISSN.2470-1173.2016.20.COLOR-307

results demonstrate that the color error is greatly decreased by employing the proposed calibrating curve III, so it is very effective to modify the trilinear interpolation model by developing effective calibrating curves. In conclusion, although the calibrating curves have only been developed and evaluated for display characterization in this article, we think that they could be applied to printer characterization[30,31] with three or more channels in future work.

ACKNOWLEDGMENTS

The authors would like to thank J. El Khoury and Xingbo Wang for their helpful suggestions, and the anonymous reviewers for constructive feedback on the manuscript. This research has been funded by China Scholarship Council, China Postdoctoral Science Foundation (No. 2014M562436) and Foundation of Shaanxi Provincial Department of Education (No. 15JK1524).

REFERENCES

[1] T. Johnson, "Methods for characterizing colour scanners and digital cameras," Displays **16**, 183–191 (1996).

[2] J. Thomas, P. Colantoni, J. Y. Hardeberg, I. Foucherot, and P. Gouton, "A geometrical approach for inverting display color-characterization models," J. Soc. Inf. Disp. **16**, 1021–1031 (2008).

[3] W. Zou, H. Xu, and R. Gong, "Efficient and accurate local model for colorimetric characterization of liquid-crystal displays," Opt. Lett. **37**, 31–33 (2012).

[4] P. Colantoni, J. Thomas, and J. Y. Hardeberg, "High-end colorimetric display characterization using an adaptive training set," J. Soc. Inf. Disp. **19**, 520–530 (2011).

[5] K. Xiao, C. Fu, D. Karatzas, and S. Wuergera, "Visual gamma correction for LCD displays," Displays **32**, 17–23 (2011).

[6] Y. Wang and H. Xu, "Colorimetric characterization of liquid crystal display using an improved two-stage model," Chin. Opt. Lett. **4**, 432–434 (2006).

[7] M. Lo, C. Chen, R. Perng, and Z. Hsieh, "The characterisation of colour printing devices via physical, numerical and LUT models," *Proc. IS&T CGIV2006: 3rd European Conf. on Colour in Graphics, Imaging, and Vision* (IS&T, Springfield, VA, 2006), pp. 95–99.

[8] H. R. Kang, *Computational Color Technology* (SPIE Press, Bellingham, WA, 2006).

[9] B. Sun, H. Liu, S. Zhou, and W. Li, "Evaluating the performance of polynomial regression method with different parameters during color characterization," Math. Probl. Eng. **2014**, 418651 (2014).

[10] P. Li, J. Wang, and J. Jing, "Application of improved back propagation algorithm in color difference detection of fabric," Color Res. Appl. **40**, 311–317 (2015).

[11] S. Bianco, F. Gasparini, and R. Schettini, "Polynomial modeling and optimization for colorimetric characterization of scanners," J. Electron. Imaging **17**, 043002 (2008).

[12] P. A. Drakopoulos and G. Subbarayan, "Color printer characterization using optimization theory and neural networks," US Patent 6,480,299 (2002).

[13] F. Yang, Y. Murakami, and M. Yamaguchi, "Digital color management in full-color holographic three-dimensional printer," Appl. Opt. **51**, 4343–4352 (2012).

[14] P. Green, J. Holm, and W. Li, "Recent developments in ICC color management," Color Res. Appl. **33**, 444–448 (2008).

[15] T. Ajito, K. Ohsawa, T. Obi, M. Yamaguchi, and N. Ohyma, "Color conversion method for multiprimary display using matrix switching," Opt. Rev. **8**, 191–197 (2001).

[16] K. Kanamori, "Interpolation errors on gray gradations caused by the three-dimensional lookup table method," J. Electron. Imaging **10**, 431–444 (2001).

[17] D. R. Wyble and R. S. Berns, "A critical review of spectral models applied to binary color printing," Color Res. Appl. **25**, 4–19 (2000).

[18] B. Sun, H. Liu, S. Zhou, C. Cao, and Y. Zheng, "Modified spectral Neugebauer model for printer characterization," Spectrosc. Lett. **48**, 660–668 (2015).

[19] H. R. Kang, *Color Technology for Electronic Imaging Devices* (SPIE Press, Bellingham, WA, 1997).

[20] R. Rossier, T. Bugnon, and R. Hersch, "Introducing ink spreading within the cellular Yule–Nielsen modified Neugebauer model," *Proc. IS&T/SID's CIC18: Eighteen Color and Imaging Conf.* (IS&T, Springfield, VA, 2010), pp. 295–300.

[21] M. E. Demichel, Procédé **26**, 17–21, 26–27 (1924).

[22] L. Yang, "Probabilistic spectral model of color halftone incorporating substrate fluorescence and interface reflections," J. Opt. Soc. Am. A **27**, 2115–2122 (2010).

[23] S. Zuffi, S. Santini, and R. Schettini, "Accounting for inks interaction in the Yule–Nielsen spectral Neugebauer model," J. Imaging Sci. Technol. **50**, 35–44 (2006).

[24] T. Bugnon and R. Hersch, "Recovering Neugebauer colorant reflectances and ink-spreading curves from printed color images," Color Res. Appl. **39**, 216–233 (2014).

[25] J. Viggiano, "Modeling the color of multicolored halftones," *TAGA Proc.* (Rochester, NY, 1990), pp. 44–62.

[26] S. Lee, R. Shamey, D. Hinks, and W. Jasper, "Development of a comprehensive visual dataset based on a CIE blue color center: assessment of color difference formulae using various statistical methods," Color Res. Appl. **36**, 27–41 (2011).

[27] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," ACM Trans. Math. Softw. **23**, 550–560 (1997).

[28] Y. Cho, H. Im, and Y. Ha, "Inverse characterization method of alternate gain-offset-gamma model for accurate color reproduction in display device," J. Imaging Sci. Technol. **50**, 139–148 (2006).

[29] E. A. Day, L. Taplin, and R. S. Berns, "Colorimetric characterization of a computer-controlled liquid crystal display," Color Res. Appl. **29**, 365–373 (2004).

[30] B. Sun, H. Liu, S. Zhou, J. Xing, C. Cao, and Y. Zheng, "A color printer calibration method based on gamut-division algorithms," Coloration Technol. **130**, 453–458 (2014).

[31] R. J. Rolleston, M. S. Maltz, and J. E. Stinehour, "Color printer calibration architecture," US Patent 5,528,386 (1996).