Monitor Characterization Model Using Multiple Non-square Matrices for Better Accuracy*

Thanh H. Ha, Satyam Srivastava, Edward J. Delp, Jan P. Allebach
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA

Abstract

We introduce a new model that consists of a color classifier followed by four non-square matrices for forward characterization of display devices (i.e. predicting the XYZ value of a displayed color when the input RGB value is known). As this forward model is not directly invertible, we present a framework for developing an optimal inverse of the forward model for characterizing display devices in the opposite direction (i.e. predicting the RGB value that should be sent to a display to obtain a known XYZ value). Finally, we discuss a method to assess the discontinuity issue of these two models on the boundaries between the color classes, and a technique which we call "overlapping training" to handle this issue. Our experimental results show that these new models outperform the conventional models with a single 3×3 or 3×10 matrix.

Introduction

The primary goal of display characterization is to develop a forward mapping from a device-dependent RGB space (or non-linear RGB space) to a device-independent color space, and vise-versa. Currently, the two most common approaches to display characterization are the lookup-table (LUT)-based method and the model-based method. The former often provides better accuracy, is easy to implement in hardware, but requires large memory and many measurements. Moreover, it is quite difficult to implement an inverse characterization process with a LUT-based approach [1,2]. On the other hand, the latter approach is simpler, does not require large memory and as many measurements as the former, but is generally less accurate. In this paper, we propose some techniques to enhance the accuracy of the model-based method. Figure 1 shows a diagram of a conventional forward model that has been widely used for CRT and LCD monitors [1]. The model consists of two main parts: a linearization module for linearizing a non-linear RGB input to a linear RGB value, and a single 3×3 matrix *M* for transforming the linear RGB value to a device-independent CIE 1931 XYZ [3] (or XYZ) value.

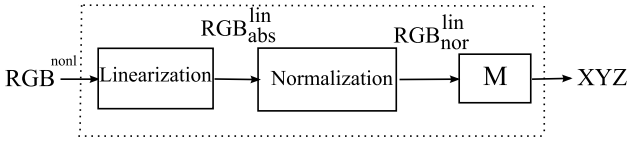

Figure 1. *The conventional forward characterization model using a single matrix. The abbreviations are $abs-$absolute, $nor-$normalized, $lin-$linear, and $nonl-$non-linear.*

Recent studies have proposed several methods for optimizing the 3×3 matrix by minimizing the transformation errors in the XYZ space, such as the Linear Regression and White-Point Preserving Least-squares (WPPLS) methods [4], or better in the 1976 CIE $L^*a^*b^*$ (or LAB) space [5], such as the Delta E Minimization (DEM) and White-point Preserving Delta E Minimization (WPDEM) methods [6]. In the WPPLS and WPDEM methods, the 3×3 matrix is constrained to guarantee that the reference white point, an important factor in color reproduction, is preserved. Although this model works quite well for many applications, for applications that require highly accurate characterization, this conventional model is not always satisfactory. In this case, we need a more complex model in order to obtain the desired level of accuracy.

As shown in [4, 6], the constraint setting for white point preservation always produces considerably higher errors for the colors that are not close to the neutral axis. This observation essentially suggests that we partition the color space into several classes, and just apply the constraint to the transformation corresponding to the class that contains the neutral colors. This approach will reduce the effect of the constraint setting while still preserving the white point of the device.

Additionally, the use of the linear 3×3 transformation is based on the assumptions that the linearization step works perfectly to linearize the non-linearity of the non-linear RGB inputs, and that there is no interaction between channels. This assumption is generally not satisfied, especially for LCD devices whose tone characteristics are quite difficult to model [1, 8, 9]. In this case, the use of a single 3×10 matrix is recommended instead [8, 9]. To further increase the accuracy of the model, we propose to use four 3×11 matrices to characterize an LCD display.

As far as we know, there has not been a clear description in the literature of how to develop the inverse for a forward characterization model that uses a non-square matrix. In this paper, we introduce a framework for developing an optimal approximate inverse for a forward model with multiple 3×11 matrices. This framework can be easily adapted to the development of the inverse for a model with a single non-square matrix too. Once the forward model is available, the development for the inverse model can be accomplished totally in software with no additional measurements.

In the two proposed models, we use different transformations for different color classes. This might cause some discontinuities on the boundaries between the neighboring color classes. In this paper, we also propose a technique which we call "overlapping training" to deal with this problem.

Throughout the paper, we will denote our new forward and inverse models with multiple 3×11 matrices as F-MM11 and I-

* This research was supported by a grant from the Indiana 21st Century Research and Technology Fund.

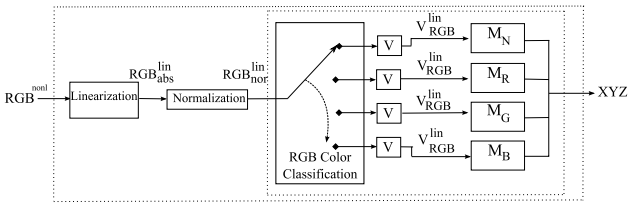

Figure 2. *Forward model with multiple 3×11 matrices (F-MM11). The inner dotted rectangle indicates what is referred to later in this paper as the core part of the F-MM11 model.*

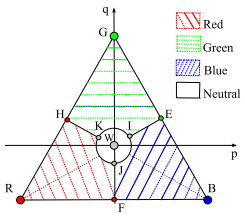

Figure 3. *The color classification in the pq diagram for the RGB^{lin} color space.*

MM11, respectively. For simplicity, unless we need to simultaneously refer to both the absolute and normalized linear RGB values, we will use the notation RGB^{lin} to refer to the latter. We will also assume that display devices use one 8-bit unsigned integer per channel to represent colors in the non-linear RGB space. The remainder of the paper is organized as follows: Next, we will describe the development of the F-MM11 and I-MM11 models. Then, we will present the experimental results for the proposed models with a particular LCD display. After that, we will discuss the overlapping training technique for smoothing the potential contour artifacts. We will conclude the paper in the final section.

Forward Model With Multiple 3×11 Matrices (F-MM11)

Figure 2 shows a block diagram of the F-MM11 model. We denote the matrices specifically developed for the four color classes: Neutral, Red, Green, and Blue as M_N, M_R, M_G, and M_B, respectively. The box V in Fig. 2 generates V_{RGB}^{lin} from RGB^{lin}. V_{RGB}^{lin} is an 11×1 vector containing the cross-product and second-order terms of the components of RGB^{lin}.

$$V_{RGB}^{lin} = [R^{lin}, G^{lin}, B^{lin}, (R^{lin})^2, (G^{lin})^2, (B^{lin})^2, R^{lin}G^{lin}, R^{lin}B^{lin}, G^{lin}B^{lin}, R^{lin}G^{lin}B^{lin}, 1]^T. \quad (1)$$

Let us describe first the method for classifying the RGB^{lin} space, and then the framework for the development of the 3×11 transformation matrix for each color class.

Color Classification in the RGB^{lin} Space

We adopt a method described in [7] to classify the RGB^{lin} space into four classes, namely Neutral, Red, Green, and Blue. Given a color described by R^{lin}, G^{lin}, and B^{lin}, the classification is performed by first projecting this color onto the chromaticity rgb diagram using

$$r = \frac{R^{lin}}{R^{lin}+G^{lin}+B^{lin}}, \quad g = \frac{G^{lin}}{R^{lin}+G^{lin}+B^{lin}}, \quad b = \frac{B^{lin}}{R^{lin}+G^{lin}+B^{lin}}. \quad (2)$$

We then transform the r, g, and b values to pq coordinates for ease of computation:

$$\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3)$$

Finally, based on the location of the color in the pq diagram, we determine the class to which the color belongs. As shown in Fig. 3, the Neutral class is bounded inside a circle centered at white point W (where $r = g = b$) with an appropriate radius. In this research, we choose a radius of 0.15 in accordance with [7]. The three other color classes are distributed symmetrically around the Neutral circle. For example, the Red class is bounded inside the lines JF, FR, RH, HK, and the arc KJ. The Green and Blue classes are defined similarly. These lines are used to determine the class to which a particular color belongs. A color on the border between two or three neighboring color classes is assigned to the class with the highest priority, according to the prioritization $N > R > G > B$.

Transformation Matrices

We develop the transformation matrix for each color class independently using a framework shown in Fig. 4. This framework is formulated as an optimization problem in which the 33 entries of M_c ($c = N$, R, G, or B) are the independent variables of a cost function that reflects the total transformation error in ΔE units[†]. In our research, we choose the root mean square error in ΔE computed over a training set as the cost function. The training set for each transformation matrix includes the samples of the corresponding color class taken from a 6×6×6 grid uniformly spanning the entire non-linear RGB cube. In order to have enough training samples for the matrix M_N, in addition to the Neutral samples in the 6×6×6 grid, we include in the training set 40 additional samples obtained by uniformly sampling the Neutral class. Among these additional samples, there are ten samples from the neutral axis. To preserve the white point, we set a constraint on the transformation matrix M_N to ensure that the white color as well as the samples on the neutral axis are transformed accurately. The other transformation matrices are optimized freely without any constraint setting. In this way, the constraint setting on M_N will not affect the accuracy of the colors in the other classes. Pattern Search [10], a strong derivative-free constrained optimization algorithm, is used for this optimization.

Inverse Model with Multiple Non-Square Matrices (I-MM11)

In this section, we will introduce a framework for developing the I-MM11 model that is the optimal approximate inverse of the non-invertible model F-MM11. A block diagram of the I-MM11 is shown in Fig. 5. Similar to the F-MM11 model, the I-MM11 model also contains a color classification module to classify the

[†] ΔE is the Euclidean metric computed in the 1976 CIE $L^*a^*b^*$ color space.

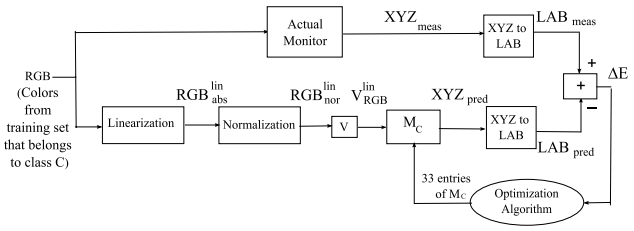

Figure 4. *The framework for developing the 3×11 matrix for a given color class c, ($c = N, R, G,$ or B). The abbreviations are: $meas$—measured and $pred$—predicted.*

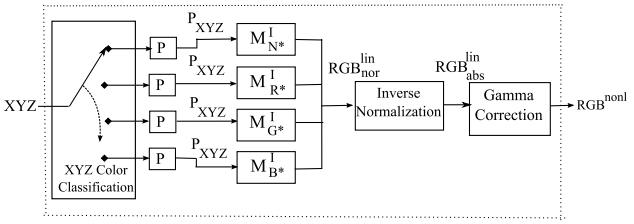

Figure 5. *Approximate inverse (I-MM11) to the forward model F-MM11 with multiple 3×11 matrices shown in Fig. 2. The gamma correction module is the inverse of the linearization step in the F-MM11 model.*

device's gamut in the XYZ space into four color classes: Neutral, Red, Green, and Blue. However, these four color classes need not be the same as the classes defined in the RGB^{lin} space. Thus, a given color RGB_0^{lin} on the input side of the F-MM11 model might map to a color XYZ_0 at the output of the F-MM11 model where RGB_0^{lin} belongs to a certain class c, but XYZ_0 belongs to a different class $c^* \neq c$. To distinguish between the color classes for the two models, we name the color classes in the XYZ space as Neutral*, Red*, Green*, and Blue*, and denote them as N^*, R^*, G^*, and B^*, respectively. We also use a non-linear 3×11 mapping between the XYZ space and the RGB^{lin} space for each class.

$$\begin{bmatrix} R^{lin} \\ G^{lin} \\ B^{lin} \end{bmatrix} = \begin{bmatrix} \\ M_{c^*}^I \\ \\ \end{bmatrix} \begin{bmatrix} \\ P_{XYZ} \\ \\ \end{bmatrix}; \qquad (4)$$

where $M_{c^*}^I$, $c^* = N^*$, R^*, G^*, or B^*, is the 3×11 transformation matrix corresponding to color class c^*, and P_{XYZ} is an 11×1 vector that contains the cross-product and second-order terms of XYZ.

$$P_{XYZ} = \left[X, Y, Z, X^2, Y^2, Z^2, XY, XZ, YZ, XYZ, 1\right]^T. \qquad (5)$$

Again, let us first describe the color classification in the XYZ space, and then the framework for developing the M_{c*}^I matrices.

Color Classification in the XYZ Space

The color classification is performed in the xy chromaticity diagram as illustrated in Fig. 6. In this diagram, R, G, and B are the three primaries which form the gamut triangle of the display. The lines RE, GF, and BH intersect at W—the reference white point. The N^* class is bounded inside a circle centered at W with an appropriate radius. In this research, we chose a radius of 0.083 in the xy diagram that approximately corresponds to the

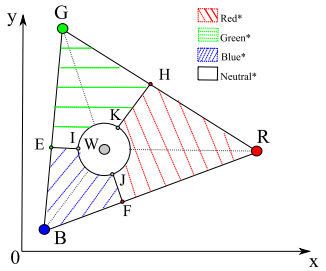

Figure 6. *The color classification in the xy chromaticity diagram for the XYZ color space.*

radius of 0.15 for the N class in the pq diagram. The regions of the other classes are very similar to those for the classification in RGB^{lin} shown in Fig. 3. For example, the R^* class is bounded inside the lines JF, FR, RH, HK, and the arc KJ. Colors that fall on the boundary between two or more classes are assigned to a class based on the same prioritization used for the classification of RGB^{lin} colors.

Transformation Matrices

To determine each matrix $M_{c^*}^I$, ($c^* = N^*$, R^*, G^*, or B^*), we formulate an optimization problem. Figure 7 shows a block diagram of the framework. In the framework, the F-MM11 model is employed to generate the training data, and also is embedded into the optimization process to train the optimal M_{c*}^I. The XYZ training values are generated by feeding the F-MM11 model a 7×7×7 grid spanning the entire RGB cube, and then classifying the generated XYZ values using the classifier in Fig. 6. Since this generation stage is implemented completely in software, we save a lot of time spent measuring the XYZ values on the actual monitor. As shown in Fig. 7, each XYZ value from the training set is directly converted to an LAB_{ref} value that will then serve as a reference. On the other hand, the vector P_{XYZ} obtained from the components of XYZ is multiplied by the appropriate matrix $M_{c^*}^I$, which is being optimized, to obtain an RGB^{lin} value. This RGB^{lin} value is put back to the core part of the F-MM11 model shown in Fig. 2 to predict the corresponding XYZ value. The predicted XYZ value is converted to LAB_{pred}, and finally compared to the reference LAB_{ref}. The optimization algorithm is used here to minimize the differences between LAB_{pred} and LAB_{ref} averaged over the training set to obtain the optimal matrices M_{c*}^I.

To preserve the white point, we set a constraint on the transformation M_{N*}^I corresponding to the N^* class so that the XYZ value of the white point is transformed exactly to the corresponding RGB^{lin} for the white point. The other matrices are optimized freely without any constraint setting.

Experiments

We compare the accuracy of the F-MM11 model with the model developed using the WPDEM method in [6], and the model with a 3×10 matrix developed using the method in [9]. We name the former the F-SM3 model and the latter the F-SM10 model. These two models are good candidates for the experiment since they have been shown to provide better results than other conventional model-based methods [6, 9]. All of the models were developed using a common training set, and tested with a common testing set. They also shared the same linearization module which

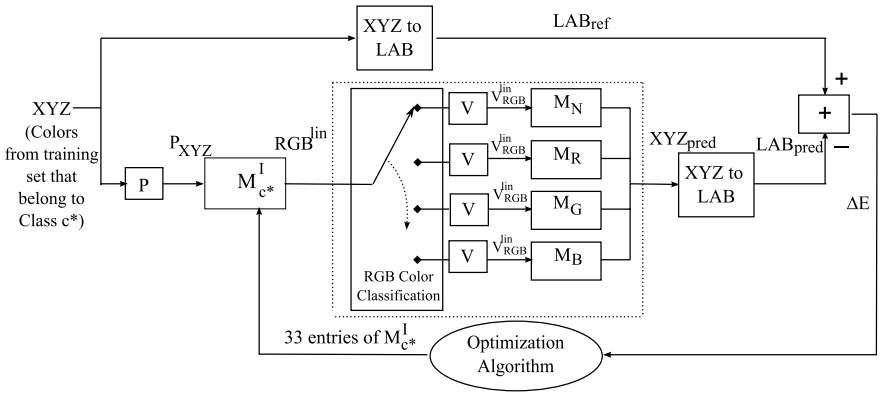

Figure 7. *The framework for determining the matrix M^I_{c*} for a given color class c^*, ($c^* = N^*$, R^*, G^*, or B^*).*

Table 1: Testing error statistics in ΔE units for the 3D-LUT and the F-SM3, F-SM10, and F-MM11 models.

Classes	Statistics	3D-LUT	F-SM3	F-SM10	F-MM11
Neutral	Mean	5.33	1.3	0.75	0.36
	Max	20.9	2.98	1.57	0.72
	Min	0.68	0.08	0.07	0.04
Red	Mean	4.76	1.43	0.89	0.54
	Max	10.59	3.32	2.4	1.12
	Min	0.84	0.07	0.05	0.02
Green	Mean	5.55	1.71	0.84	0.49
	Max	15.75	3.35	1.87	1.2
	Min	1.28	0.11	0.09	0.1
Blue	Mean	6.85	2.78	0.89	0.53
	Max	14.91	6.41	1.75	1.39
	Min	1.83	0.61	0.11	0.10

Table 2: Testing error statistics in ΔE units for the I-SM3 and I-MM11 models.

Classes	Statistics	I-SM3	I-MM11
Neutral*	Mean	1.47	0.64
	Max	5.94	2.01
	Min	0.21	0.17
Red*	Mean	1.67	0.65
	Max	3.50	1.5
	Min	0.13	0.08
Green*	Mean	1.53	0.88
	Max	2.76	1.74
	Min	0.26	0.18
Blue*	Mean	1.78	0.63
	Max	3.74	1.54
	Min	0.35	0.03

was accomplished by applying three 1D-LUTs independently to the three channels of the display. Each 1D-LUT was populated by interpolating the measured luminance values of 64 digital levels of the corresponding channel. The training set included the samples from a 6×6×6 grid in the RGB cube augmented with 40 Neutral samples. The testing set included the samples from a 5×5×5 uniform grid in the RGB cube augmented with 43 Neutral samples. No color was common to both the training and testing sets. As a further test, we also built a six-node 3D-LUT (6×6×6) using trilinear interpolation [2]. The target display device was a Dell UltraSharp 2408WFP 24-inch LCD display[‡]. Table 1 summarizes the testing error statistics in ΔE units for these three models and the 3D-LUT. This shows the improvements obtained with the F-MM11 model. It also shows that when we need very high accuracy with a limited number of training points, the F-MM11 model is a better candidate than a 3D-LUT-based method.

Next, we compare how the inverse models performed to characterize this LCD display in the reverse direction. As aforementioned in the "Introduction" section, inverse characterization with a 3D-LUT is quite difficult. Also, [9] does not describe how to develop the inverse model corresponding to the F-SM10 model. Consequently, in this experiment, we only compare the I-MM11 model with the inverse of the F-SM3 model, denoted as the I-SM3 model. Since the F-SM3 model is invertible, the I-SM3 model was obtained simply by inverting the components of the F-SM3 model. The testing set was the same as that used for testing the forward models. For each color in the testing set, we used the following procedure to test the inverse characterization of the two models:

1. Send RGB_1 to the actual monitor, measuring its XYZ value (denoted as XYZ_1).
2. Use the inverse model to predict RGB_2 from XYZ_1. If the model works perfectly, $RGB_2 = RGB_1$.
3. Send RGB_2 to the actual monitor, measuring its XYZ value (denoted as XYZ_2).
4. Compute the difference in ΔE units between XYZ_1 and XYZ_2.

Table 2 summarizes the error statistics of the two models. It shows that the I-MM11 model outperformed the I-SM3 model in the inverse characterization process.

Discontinuity Issue

Assessment of the Discontinuity

Since the F-MM11 and I-MM11 models use different transformation matrices for the different color classes, there may be

[‡] Dell Inc. One Dell Way, Round Rock, Texas, USA.

some discontinuities in the transformation on the boundaries between the neighboring color classes. To determine the extent to which a characterization model is discontinuous, we use the following technique: Let $f_1()$ and $f_2()$ be two continuous transformations developed specifically for Class 1 and Class 2, respectively. In the domain space, assume x_0 is a color of Class 1 and lies on the border between Class 1 and Class 2. Let S be a set of neighboring colors of x_0 in Class 1; and T be a set of the neighboring colors of x_0 in Class 2. If the transition at the boundary between $f_1()$ and $f_2()$ is smooth, we expect that the difference Δ computed using (6) should be close to or smaller than a visual limit. In other words, if the transition is smooth, then with a small change in the domain space, we will obtain a correspondingly small change in the range space. To gain a sense of how small it should be in the range space, we want it to be not too far away from the maximum change we obtain when using only the continuous function $f_1()$ for both classes. This guarantees that the behavior of the model using a combination of two different functions $f_1()$ and $f_2()$ is similar to that of a model using only $f_1()$. Thus, we define

$$\begin{aligned} d_1 &= \max\left(\|f_1(x_0) - f_1(x_i)\|\right);\ \forall x_i \in S \cup T, \\ d_2 &= \max\left(\|f_1(x_0) - f_2(x_j)\|\right);\ \forall x_j \in T, \\ \Delta &= |d1 - d2|; \end{aligned} \tag{6}$$

where $|\circ|$ denotes absolute value, and $\|\circ\|$ is a color difference metric.

As an example, we employed this technique to test the discontinuity of the F-MM11 model developed in the previous section. The domain of the F-MM11 model is the non-linear RGB space, and the range is the XYZ space. In the non-linear RGB space, we consider two RGB_1 and RGB_2 values to be neighboring if their components differ from each other by an amount no bigger than 1 unit−the smallest difference in the discrete RGB color space. To obtain the non-linear RGB values of the colors on the border between two classes, we first uniformly sample 30 points along their border on the pq diagram. We then convert these points from the pq diagram to the rgb diagram using

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix}. \tag{7}$$

Corresponding to each converted point on the rgb diagram, we uniformly sample the ray passing through that converted point and the origin of the RGB^{lin} coordinates to obtain 50 samples inside the RGB^{lin} cube. Doing this for the 30 converted points on the rgb diagram, we will obtain 1500 colors for each pair of color classes. Finally, we transform these colors to the non-linear RGB space. Each of these colors will serve as x_0 in the aforementioned test. For each x_0, we classify its neighbors in the non-linear RGB space to obtain the sets S and T. Using (6), we compute Δ in ΔE units for each x_0. Table 3 reports the maximum Δ value corresponding to each pair of color classes.

From Table 3, we can see that on the boundaries between the pairs Neutral-Red, and Red-Blue, there exist some pairs of colors that could potentially cause contour artifacts (max Δ values are near two ΔE units). They need to be smoothed out.

In the case of the I-MM11 model, the domain is the XYZ space and the range is the non-linear RGB space. To avoid the effects of the quantization errors, the non-linear RGB space should not be quantized for this experiment. Since the test for the I-MM11 model is very similar to that for the F-MM11 model which we have just described, we will not discuss it further here. Instead, we will move right away to the discussion about the smoothing technique.

Table 3: Discontinuity assessment on the boundaries between the color classes for the F-MM11 model. (Δ is computed in ΔE units using (6)).

Pair of Classes	Max Δ
Neutral-Red	1.92
Neutral-Green	0.65
Neutral-Blue	0.57
Red-Green	1.24
Red-Blue	1.70
Green-Blue	0.76

Smoothing Technique

In addition to the solid borders between the neighboring classes shown in Fig. 3 for the F-MM11 model, or Fig. 6 for the I-MM11 model, we define "boundary regions" for each color class. A color class, say Blue, has three types of boundary regions: Type I (with Green or Red), Type II (with Neutral), and Type III (with both Neutral and Green, or with Neutral and Red) as shown in Fig. 8. However, the Neutral class has only Type II and Type III boundary regions.

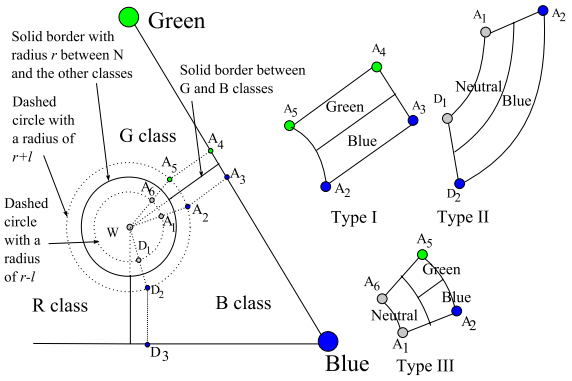

Figure 8. *Three types of boundary regions between the Blue class and the Neutral and Green classes. Assume the circle centered at the white point W bounding the Neutral class has a radius of r. Then, the two dashed circles centered at W in the figure have the radii of $r+l$ and $r-l$, respectively, where l is a threshold used to define the boundary regions. A_i, ($i = 1,2,3,4,5,6$), and D_j, ($j = 1,2,3$) are the intersection points as shown in the figure. The lines A_2A_3 and A_4A_5 are parallel to the solid border between the Green and Blue classes. The distance between each line and this solid border is l. The Type I boundary region between the Blue and Green classes is the area bounded by the lines A_2A_3, A_3A_4, A_4A_5, and the arc A_5A_2. The Type II boundary region between the Blue and Neutral classes is the area bounded by the arcs A_1D_1, A_2D_2, and the lines A_1A_2, D_1D_2. The Type III boundary region between the Blue class and the Neutral and Green classes is the area bounded by the arcs A_1A_6, A_2A_5, and the lines A_1A_2, A_5A_6.*

The smoothing technique is described as follows: In the

Table 4: Assessment of the discontinuity between the Neutral and Red classes before and after applying the overlapping training technique.

		Before	After
Pair of Classes	Statistics	Δ (in ΔE)	Δ (in ΔE)
Neutral-Red	Mean	0.36	0.31
	Max	1.92	0.62
	Min	0.0001	0.001

Table 5: Testing error statistics in ΔE units for the F-MM11 model in the Neutral and Red classes before and after applying the overlapping training technique.

Classes	Statistics	Before	After
		Error (ΔE)	Error (ΔE)
Neutral	Mean	0.36	0.35
	Max	0.72	0.68
	Min	0.04	0.07
Red	Mean	0.54	0.53
	Max	1.12	1.07
	Min	0.05	0.18

training stage for the transformation matrix for a particular color class, say Blue, instead of using only the training samples in the Blue class itself, we include in the training set the samples of the Neutral, Red, and Green classes that are in the boundary regions with the Blue class.

As shown in Table 3, for the F-MM11 model, the discontinuity is most severe in the boundary region between the Red and Neutral classes. As an example, we apply the smoothing technique to this pair of color classes. We choose a threshold l of 0.06, an arbitrary small number in the pq coordinates, to define the boundary region between the Neutral and Red classes. Table 4 shows the statistics of Δ computed by (6) before and after applying the smoothing technique, using the same set of testing points. As shown in Table 4, the maximum Δ value after applying the smoothing technique is reduced to 0.62 ΔE units, an acceptable value. As the space of each class is extended, one might expect that the characterization errors outside the boundary regions would increase. Table 5 shows the comparison between the testing error statistics for the F-MM11 model in the Neutral and Red classes before and after applying the smoothing technique. The before-smoothing-statistics are the same as those in the last column of Table 1 for these two color classes. We see that although the minimum errors do increase, the mean and maximum errors actually decrease. This is not entirely surprising because we would expect a more accurate transformation for colors in the boundary region and a less accurate transformation for colors in the center of each class region. Thus, there is clearly a trade-off that makes the overall outcome difficult to predict.

Conclusions

In this paper, we have introduced a new forward model (F-MM11) and a new inverse model (I-MM11), each of which consists of a color classifier followed by four non-square matrices, for display characterization. Our experimental results show that both new models outperform in accuracy other previously reported model-based methods. The framework which we have described for developing the I-MM11 model is easy to implement. This framework is also applicable to the development of the optimal inverse for other non-invertible models with a single non-square matrix. We have also discussed a method to assess the discontinuity of these new models on the boundaries of color classes, and proposed an overlapping training technique to handle this issue. The results show that the overlapping training technique works quite well to handle the boundary issue. Although our proposed models are more complex than the conventional models with a single matrix, the high accuracy that they can achieve without many measurements makes them very useful in applications that require highly accurate characterization.

References

[1] B. Cressman, B. Bastani, and B. Funt, "Calibrated colour mappping between LCD and CRT Displays: a case study," in *Proceeding of CGIV 2004: The Second European Conference on Colour Graphics, Imaging and Vision*, pg. 57–61 (April 2004).

[2] H. Kang, *Computational Color Technology*, SPIE Publications, Washington, USA (2006).

[3] CIE (1932), *Commission internationale de l'Eclairage proceedings*, Cambridge University Press, Cambridge (1931).

[4] G. Finlayson and M. Drew, "White-point preserving color correction," in *Proceeding of IS&T/SID 5th Color Imaging Conference: Color Science, Systems and Applications*, pg. 258–261 (November 1997).

[5] CIE. Colorimetry, *CIE Publication No. 15.2*, Central Bureau of the CIE, Vienna (1986).

[6] B. Funt, R. Ghaffari, and B. Bastani, "Optimal linear RGB to XYZ mapping for color display calibration," in *Proceeding of IS&T/SID 12th Color Imaging Conference: Color Science, Systems and Applications*, pg. 223–227 (November 2004).

[7] H. Haneishi, K. Miyata, H. Yaguchi, and Y. MiYake, "A new method for color correction in hardcopy from CRT images," *Journal of Imaging Science and Technology* **37**(1), pg. 30–36 (1993).

[8] P. Bodrogi and J. Schanda, "Testing a calibration method for colour CRT. A method to characterize the extent of spatial interdependence and channel interdependence," *Displays* **16**, pg. 123–133 (1995).

[9] O. Arslan, Z. Pizlo, and J. Allebach, "CRT calibration techniques for better accuracy including low-luminance colors," in *Proceeding of Color Imaging IX: Processing, Hardcopy, and Applications. Edited by Eschbach, Reiner; Marcu, Gabriel G. (Proceedings of the SPIE)*, pg. 286–297 (January 2003).

[10] R. Lewis and V. Torczon, "Pattern search methods for linearly constrained minimization," *SIAM Journal on Optimization* **10**(3), pg. 917–941 (2000).

Author Biography

Thanh Huy Ha has been a PhD student in the School of Electrical and Computer Engineering at Purdue University since 2005. Before joining the PhD program at Purdue University, he worked as a Researcher at the National Center of Technology Progress in Hanoi, Vietnam. He holds a B.S degree in Electronics and Telecommunications from Hanoi University of Technology. His research interests include image analysis and color management.