Calibrated Colour Mapping Between LCD and CRT Displays: A Case Study

Bill Cressman, Behnam Bastani, and Brian Funt
Simon Fraser University
Burnaby BC, Canada

Abstract

The primary goal of a colour characterization model is to establish a mapping from digital input values d_i (i=R,G,B) to tristimulus values such as XYZ. A good characterization model should be fast, use a small amount of data, and allow for *backward* mapping from tristimulus to d_i. This paper demonstrates implementations of three different colour characterization models, each tested on seven display devices. The characterization models implemented in this study are a 3D LUT, a linear model,[2] and the masking model introduced by Tamura et al. in 2002.[6] The devices include two CRT Monitors, three LCD Monitors, and two LCD Projectors.

Several characteristics of the display devices are presented in relation to data collection and characterization modeling. These include the long phosphor stabilization time on CRT monitors and the shifting chromaticity of mixed colours on LCD displays.

The results of this study indicate that a simple linear model is the most effective for all devices used in the study, despite the common belief that it is sometimes inappropriate for LCD monitors. A simple extension to the linear model is presented, and it is demonstrated that this extension improves white prediction without causing significant errors for other colours.

Introduction

Accurate colour management across multiple displays is an important problem, and will become more important in years to come. Users are increasingly relying on digital displays for creating, viewing and presenting colour media. Users with multi-panel displays would like to see colour consistency across the displays, while conference speakers would like an accurate prediction of what their slides will look like before they enter the auditorium.

The act of predicting colours across multiple display devices requires implementation of several concepts, including device characterization, gamut mapping, and perceptual models. This paper is focused on the concept of device characterization – establishing a mapping from digital input values d_i (i=R,G,B) to tristimulus values such as XYZ. A good characterization model should be fast, use a small amount of data, and allow for *backward* mapping from tristimulus to d_i.

There are a several well-known characterization models that support both forward and backward mapping, three of which were implemented in this experiment: 3D Lookup Table (LUT), linear model and masking model. The 3D LUT method uses a pair of three-dimensional tables to associate a tristimulus triplet with every RGB combination, and vice versa. This method is simple to understand, but difficult and cumbersome to implement.

The term *linear model* refers to the group of models (GOG, S-Curve, and Polynomial model) that estimate tristimulus response with a linear combination of pure phosphor output. These models each start by linearizing the digital input response curves with the specific nonlinear function from which they draw their names. The linear model has been widely used for CRT monitors, but has been criticized for its assumption of channel independence, which may not apply on LCD displays.

The third model implemented in this study was the masking model introduced by Tamura et al. in 2002.[6] This model applies the concept of Under Colour Removal (UCR) to mask inputs from 3-dimentional RGB space to 7-dimensional RGBCMYK space, then linearizes inputs and combines outputs as was done in the linear model.

This paper will discuss the implementation, benefits, and pitfalls of each method with respect to use on CRT and LCD display devices. In general, prediction errors will be quantified terms of ΔE, as measured in 1994 CIE La*b* colour space. The first section of the paper deals with data collection. The next section reviews the characteristics of devices used in the study. Section 3 discusses implementation details and considerations for each of the three characterization models, and section 4 reviews the results of the study and proposes improvements.

Data Collection

All data used in this study was collected using a Photo Research SpectraScan 650 Spectrometer in a dark room with the spectrometer at a fixed distance, perpendicular to the center of the display surface. Before beginning each test, the monitor settings were re-set to the factory default and the brightness was adjusted using a gray-scale calibration pattern until all shades of gray were visible.

The data collection was performed automatically in large randomized test suites. We found that it is important to test the repeatability of the spectrometer with respect to each monitor and ensure that the test plan is sufficient to smooth out significant measurement errors. As a result, each RGB sample used in this study was composed of a total of 25 measurements, taken in 5 randomly scheduled bursts of 5 measurements each.

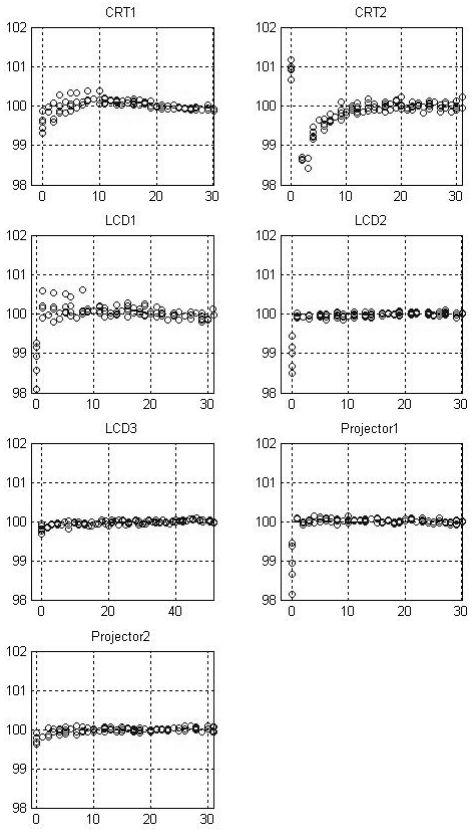

Figure 1. Percentage of the steady state luminosity for white on the vertical-axis vs. the number of seconds since black was displayed on the horizontal-axis.

Figure 1 shows the percentage of steady state luminosity for white vs. the number of seconds since a colour change from black. In this paper, luminosity will be defined as the L value in $CIELAB_{94}$ space. Note that the LCD-based devices often reach steady state within less the first second, while the CRT devices take longer. The amount of time required for the CRT devices was somewhat surprising – up to 10 seconds in CRT2. The spike that occurs on CRT2 right after the colour change is unexpected as well. However, the implication for testing is straightforward - measurement delay after a colour change must be several seconds longer for CRT devices.

Another important setting related to data consistency is spectrometer integration time. In general, CRT monitors require a longer integration time because the display flashes with each beam scan. Figure 2 shows the result of an integration time test on CRT1.

Observe that shorter lower integration times result in more unstable measurements. The monitor refresh rate used in this experiment is 75 Hz, which equates to 13.3 ms per scan. Therefore, any integration time t will experience either $\lfloor t/13.3 \rfloor$ or $\lceil t/13.3 \rceil$ scans depending on when the measurement window starts. For example, if the integration time is 100ms, then measurements will one, either experience seven or eight scans, leading to high variation. Conversely, a time of 400 ms will almost always lead to 30 scans (400 / 13.33 = 30.00).

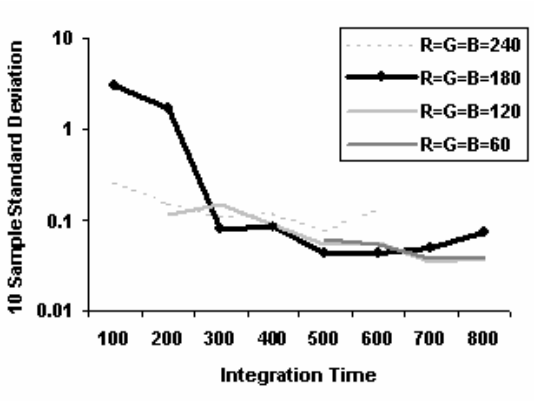

Figure 2. Measurement Error (Log scale) vs Integration Time in milliseconds measured on four grayscale colours on CRT1

The measurements in this study were taken with a default integration time of 400ms, which was doubled whenever a "low light" error was detected and halved when a "too much light" error was detected. Although this technique resulted in acceptable error levels, an improvement would be to use only integration times are exact multiples of 13.3.

Three suites of data were collected for each monitor: a 10x10x10 grid of evenly spaced RGB values covering the entire 3D space, a similar 8x8x8 grid used for testing and verification, and a "101x7" data set made up of 101 evenly spaced measurements for each primary RGB and secondary CMYK channel with the other inputs set to zero.

Device Characteristics

Seven devices were tested – two CRT monitors, three LCD monitors, and two LCD projectors. A summary of these devices is given in Table 1.

Table 1. Device Summary

Name	Description
CRT1	Samsung Syncmaster 900NF
CRT2	NEC Accusync 95F
LCD1	IBM 9495
LCD2	NEC 1700V
LCD3	Samsung 171N
PR1	Proxima LCD Desktop Projector 9250
PR2	Proxima LCD Ultralight LX

A common issue in device characterization is channel interaction. In this study, channel interaction is calculated as follows, where v represents the input value for the channel in question, a and b are constant values for the other two channels, and $L(r,g,b)$ represents the measured luminosity for a given digital input.

$$CI_{RED}(v,a,b) = \frac{(L(v,a,b) - L(0,a,b)) - (L(v,0,0) - L(0,0,0))}{L(255,255,255) - L(0,0,0)} \quad (1)$$

This equation returns zero when there is no channel interaction. The equations for CI_{GREEN} and CI_{BLUE} are similar.

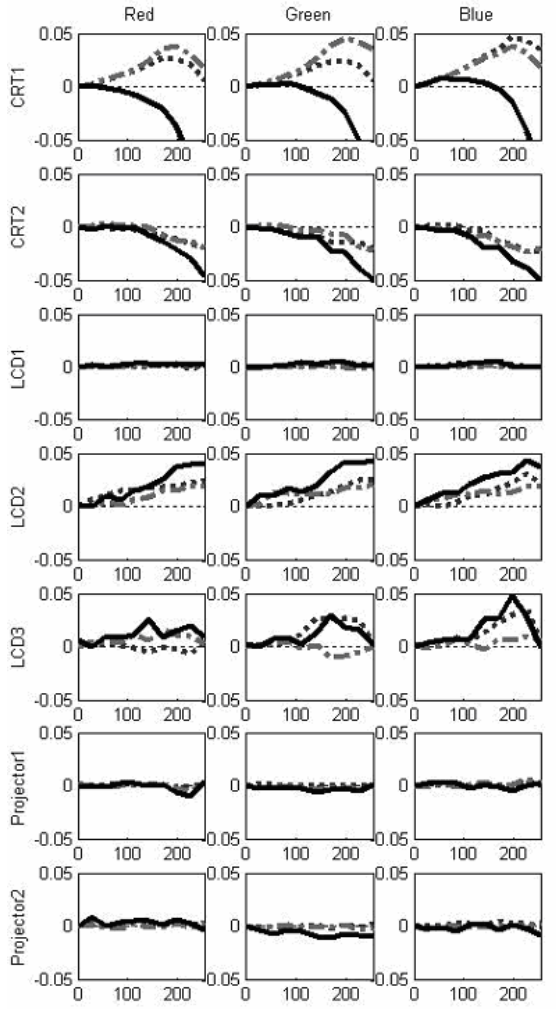

Figure 3. Channel Interaction Interaction. The horizontal axis represents the input value v ranging from 0 to 255 and the vertical axis represents the value of the Channel Interaction metric. $CI_{COLOR}(v,a,b)$. The black line shows a=b=255 and dashed lines show a=0,b=255 and a=255,b=0

It is commonly expected that LCD devices will exhibit channel interaction and CRT devices will not. However, the two CRT monitors exhibited more significant interaction problems than three of the five LCD devices, as shown in Figure 3.

The nature of the interaction is surprising as well. Observe that for CRT1, interactions with one other phosphor tend to increase luminosity output while interactions with both other phosphors tend to decrease luminosity output. Interactions on other devices were either consistently additive or subtractive.

Another potential issue with LCD monitors is chromaticity shift.[6] This study found that chromaticity shift of pure phosphor colours was insignificant. However, chromaticity shift of combined colours (CMYK) was notable on all LCD devices (Figure 4).

This effect is caused by the dissimilarity of shape between the strongly s-shaped B response curves and the more gamma-shaped R and G curves. An example of this shape difference is given in Figure 5.

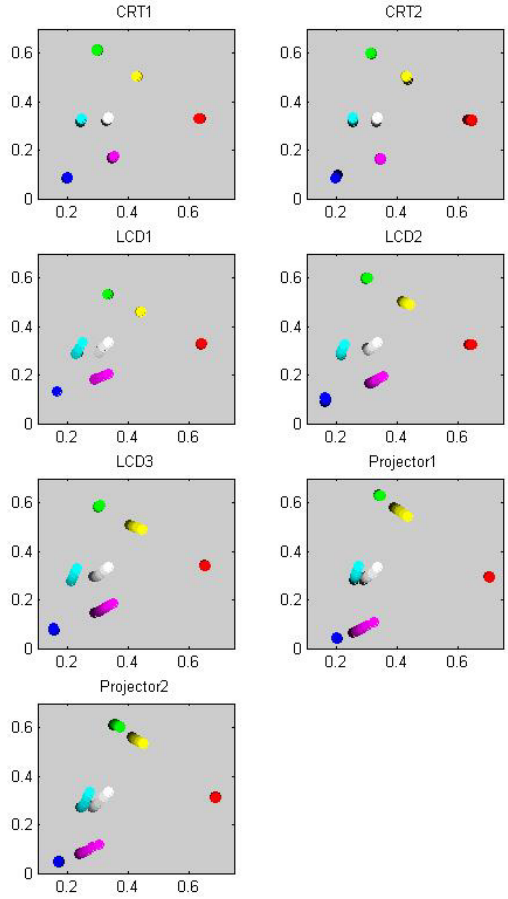

Figure 4. Chromaticity Shift Diagrams in xy space, with x=X/(X+Y+Z) on the horizontal axis and y=Y/(X+Y+Z) on the vertical axis.

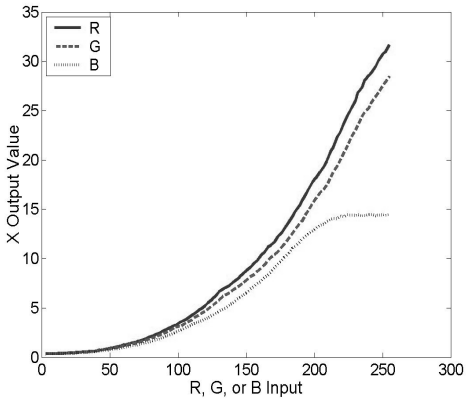

Figure 5. X response shape differences between channels for PR1

Implementation Details

All characterization methods start with black level flare correction, in which the measured XYZ value of black for the device is subtracted from the measured tristimulus value of each colour. This ensures that all devices have a common black point of (0,0,0) in XYZ space.[2] The remaining steps for each characterization are described below.

3D LUT Model

The 3D LUT method was implemented with the intention of providing a golden standard against which to evaluate the other two models, but is expensive both computationally and storage-wise (10 MB for a storage table) and is not well suited for reverse mapping. To create the forward lookup table, the 10x10x10 training data was interpolated using 3D linear interpolation to fill a 52x52x52 lookup table indexed by RGB values spaced 5 units apart. At look-up time, 3D spline interpolation is used to look up intermediate values.

Inverting the lookup to index by XYZ is non trivial – it requires interpolation of a sparse 3D data set; a task that is a field of research in its own right.[5] The reverse lookup was performed via tetrahedral interpolation on the original 10x10x10 data set. Tetrahedral Interpolation was chosen over a number of other methods primarily for its speed and ability to handle sparse, irregularly spaced data. However, any values that fall outside the convex hull of the measured gamut will return errors. This is particularly problematic for the LCD monitors, which have slightly convex gamut faces. In order to prevent edge values from returning invalid data, the entire lookup table was expanded outward by 1% from the gamut centroid.

Linear Model

The linear model is a two-stage characterization process. In the first step, the raw inputs d_i (i=1, 2, 3 for R, G, B) are linearized using a fitted function $C_i(d_i)$ for each channel. Linear regression is then used to determine the slope M_{ij} between each linearized input $C_i(d_i)$ and the respective XYZ outputs where j=(1, 2, 3) for (X, Y, Z). The second stage applies matrix M to calculate estimated XYZ values.

$$\begin{bmatrix} X_{est} \\ Y_{est} \\ Z_{est} \end{bmatrix} = M \begin{bmatrix} C_1(d_1) \\ C_2(d_2) \\ C_3(d_3) \end{bmatrix} \quad (2)$$

The linearization functions in this implementation avoid any shape predisposition by using a LUT that is calculated as follows. The 10 measured response values for each input channel i are interpolated to obtain three output vectors $X(d_i)$, $Y(d_i)$ and $Z(d_i)$ in 256-dimensional space. Principal component analysis is then used to find the single vector $C_i(d_i)$ that best approximates all three output vectors. The following equation calculates $C_i(d_i)$ where PCA_i represents the weighting vector obtained from principal component analysis.

$$C_i(d_i) = \begin{bmatrix} X(d_i) & Y(d_i) & Z(d_i) \end{bmatrix} * \begin{bmatrix} PCA_i \end{bmatrix} \quad (3)$$

In order to allow for backward mapping, two conditions are required: the linearization function must be monotonic and the matrix M must be invertible. Inversion is always possible because none of the input channels are linearly dependent. However, the monotonicity requirement is a real risk with LCD displays, where the response curves sometimes level out or even decline for high input values (Figure 6). It is therefore necessary to modify the linearization function to ensure monotonicity. Note that this modification, although necessary, serves to reduce the accuracy of the linearization and increases the overall error of the characterization.

When creating the lookup table, a decision must be made regarding the size of the training data set. Figure 7 shows the relationship between training data size and forward mapping error, measured in ΔE. In general, a larger training set is better, but the benefit tapers off after about 10 data points. For the results section of this paper, a training data set with 101 points was used to ensure minimal error introduced by training data size.

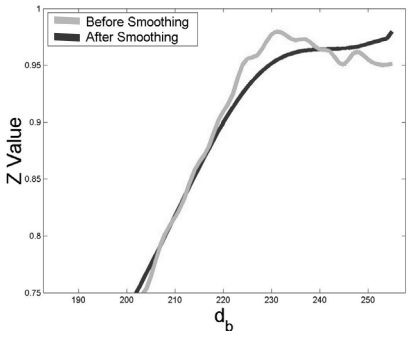

Figure 6. Smoothing correction for non-monotonicity in the Z-response curve of the B channel for PR1

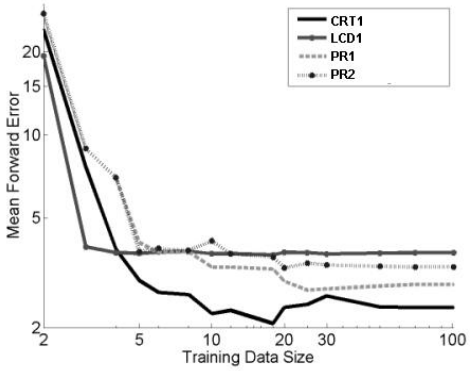

Figure 7. Mapping Error vs Training Data Size

The primary criticism of the linear model is that it assumes channel independence. As we have seen above, this is not always a valid assumption – even for CRT monitors. When there is channel interaction, the predicted output for colours that use more than one phosphor may not be accurate. This is especially true for white, which uses the maximum value of all three phosphors. Our observations suggest that this problem is not very

noticeable on natural images where the eye is accustomed to correcting for scene lighting. However it becomes significant on computer-generated images such as presentation slides or charts where there are large regions of pure white with no expected ambient lighting. In this case, the eye is less forgiving.

One solution is to perform a white-point correction to ensure that the predicted white exactly matches the measured white. A simple approach is to apply a diagonal transform to the slope matrix M based on the measured and predicted values of pure white. The following formula shows the conversion, where $X_{MEASURED}$ is the measured X value for white and $X_{PREDICTED}$ is the predicted X value for white using the original slope matrix.

$$M_{LINEAR+} = M * \begin{bmatrix} \frac{X_{MEASURED}}{X_{PREDICTED}} & 0 & 0 \\ 0 & \frac{Y_{MEASURED}}{Y_{PREDICTED}} & 0 \\ 0 & 0 & \frac{Z_{MEASURED}}{Z_{PREDICTED}} \end{bmatrix} \quad (4)$$

This modification to the slope matrix ensures that predicted white is correct, but slightly shifts all of the other colours in a non-uniform manner, which could potentially increase the overall error. This model will be referred to as "Linear+" in this paper, and is useful when displaying computer-generated images where white is a major colour. Note that a similar correction can be performed using an alternate tristimulus space, such as LMS. In our study, we found that using either XYZ or LMS intermediate space returns the same average increase in forward error (±0.05 ΔE).

Further improvement may be possible using a technique similar to that presented by Finlayson and Drew in Ref. [3], where a modified least-squares procedure is used to determine the matrix M. By constraining the prediction error for white to zero, a matrix can be selected that reduces overall error while ensuring an accurate white value. It is interesting to note that their approach achieved good results even without first linearizing the inputs.

Masking Model

The masking model[6] attempts to avoid problems related to channel interaction with a technique similar to UCR in printing. The original digital input d_i is converted to masked input m_i (i=1,2,3,4,5,6,7 for RGBCMYK), and the masked values are combined in a manner similar to what was done in the linear model. The masking operation assigns values to three elements of m – the primary colour (index p), the secondary colour (index s), and the gray colour (index 7), and sets all of the remaining elements of m to zero, as follows.

$$\begin{aligned} \text{Primary index } p \text{ such that } d_p &= \max(d_1, d_2, d_3) \\ \text{Gray index } k \text{ such that } d_k &= \min(d_1, d_2, d_3) \;\&\; k \neq p \\ \text{Secondary color index s} &= k + 3 \\ \text{Primary color value } \; m_p &= d_p \\ \text{Secondary color value } \; m_s &= d_{6-p-k} \\ \text{Gray (Under) color value } \; m_7 &= d_k \\ \text{Unused color values } m_q &= 0 : q \notin \{p, s, 7\} \end{aligned} \quad (5)$$

The result of these formulas is to set p to the index of the maximum primary colour (R, G, or B), and m_p to the input value for that colour. It assigns s to the index of the mixed colour (C, M, or Y) that does not contain the minimum colour, and assigns m_s to the median of the original values. Finally, it sets the gray value m_7 to the minimum of the three original inputs. For example, if the original inputs are RGB=(200,100,50), the primary colour will be red, with a value of 200. The secondary colour will be yellow (which does not contain blue) with a value of 100, and the gray (under) colour will have a value of 50. The masked input array becomes m=[200,0,0,100,0,0,50].

Once the inputs have been converted into masked values m_i, a linearization function $C_i(m_i)$ for each input channel i is determined using the method described above for the linear model. The slope matrix M_{ij} for each input channel i and output channel j is calculated as using PCA and linear regression, also as described for the linear model. Finally, let the vector P_i represent the column of matrix M that contains the X, Y, and Z slopes for input channel i. The transformation from masked input to XYZ output can then be written as follows:

$$XYZ_{est} = \begin{bmatrix} P_p & P_s & P_7 \end{bmatrix} * \begin{bmatrix} C_p(m_p) - C_p(m_s) \\ C_s(m_s) - C_7(m_7) \\ C_7(m_7) \end{bmatrix} \quad (6)$$

The inverse mapping from XYZ to RGB is less obvious, and requires knowledge of the primary and secondary colour indices p and s. There is no way to know these values, so all six possible (p, s) combinations are tested (RM, RY, GC, GY, BC, BM) and any combination that satisfies the following conditions will yield the correct result.

$$255 \geq m_p \geq m_s \geq m_7 \geq 0 \quad (7)$$

Results

This study calculated values of forward error ΔE_{FWD}, round trip error ΔE_{TRIP}, and backward error ΔE_{BWD} for 512 colours in an 8x8x8 evenly spaced grid of RGB inputs. For each colour, we find three vertices in CIE L*a*b* space: the measured value for the colour v_M, the predicted value v_P, and a round-trip value v_{RT} found by mapping backward and forward again from v_P. These points form a triangle with edges representing the forward, round-trip and backward error vectors. ΔE_{FWD} is the distance from v_M to v_P, ΔE_{TRIP} is the distance from v_P to v_{RT}, and ΔE_{BWD} is the distance from v_{RT} back to v_M.

Table 2. Mean Forward Errors (ΔE_{ab})

	LUT	Lin	Lin+	Mask
CRT1	0.8	2.4	2.2	2.6
CRT2	0.5	1.7	2.7	1.5
LCD1	0.8	0.9	0.9	3.5
LCD2	0.9	3.1	3.2	3.3
LCD3	1.0	3.7	3.7	4.2
PR1	1.4	1.7	1.9	5.6
PR2	0.3	2.1	2.6	7.3
Average	0.8	2.2	2.5	4.0

Table 3. Mean Backward Errors (ΔE)

	LUT	Lin	Lin+	Mask
CRT1	1.5	2.4	2.2	2.6
CRT2	1.6	1.7	2.7	1.5
LCD1	1.8	0.9	0.9	3.5
LCD2	2.4	3.1	3.2	3.3
LCD3	2.7	3.7	3.7	4.2
PR1	2.8	1.7	1.9	5.6
PR2	1.9	2.1	2.6	7.3
Average	2.1	2.2	2.5	4.0

With respect to forward or backward error, we see that the 3D LUT is the most accurate, followed by the linear, Linear+ and Masking models (Table 2, Table 3). A comparison of backward error distributions (Figure 8) shows that the linear model had tightest distribution for each device, while the distribution for 3D LUT tended to have a number of high-error outliers. The cause of these outliers becomes apparent when the error values are plotted by chromaticity. Figure 9 shows the chromaticity coordinates for points that are greater than half the maximum error for each model/device combination. Observe that the largest errors for the 3D LUT are often on or near the gamut boundary, which is where the tetrahedral interpolation tends to fall apart.

For the linear model, the highest errors are fairly well distributed across the chromaticity space for all devices except the projectors, which have a distinct problem in the blue region. This is most likely due to the non-monoticity exhibited by the projectors in the blue output curves. As mentioned in the implementation section, the monotonicity correction stage is a potential source of error for all devices. However, it appears to be adding very little error for devices that do not have a monotonicity problem (Table 4). The most notable increase in error was seen with the Projector 1, which also had the most trouble with non-monotonicity.

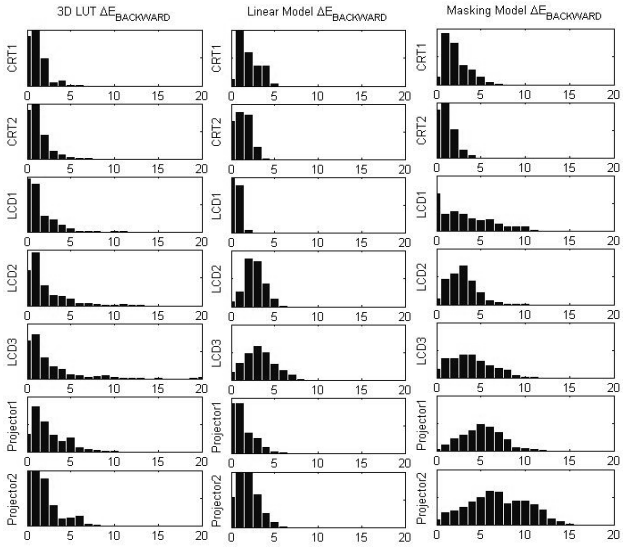

Figure 8. Backward Error distribution for each characterization model on each device. ΔE error value is shown on the horizontal axis and histogram counts are shown on the vertical axis

Table 4. Percent Increase in Forward ΔE Error Due to Monotonicity Correction

	Uncorrected	Corrected	% Increase
CRT1	2.4	2.4	0.0%
CRT2	1.7	1.7	0.0%
LCD1	0.9	0.9	-0.7%
LCD2	3.1	3.1	0.0%
LCD3	3.5	3.7	4.7%
PR1	1.5	1.7	12.4%
PR2	2.1	2.1	-0.8%
Average	2.2	2.2	2.3%

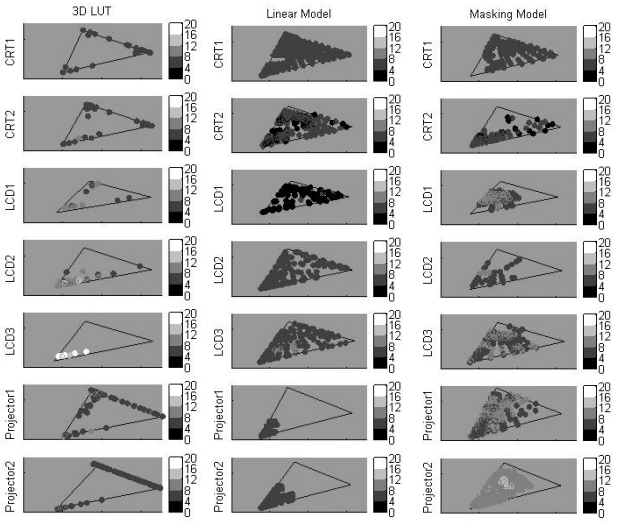

Figure 9. Backward Error vs Chromaticity. The horizontal and vertical axes are X/(X+Y+Z) and Y/(X+Y+Z) respectively.

The results in Table 2 and Table 3 show that the average error for the Linear+ model was nearly the same as that for the standard linear model. Recall that the goal of Linear+ is to guarantee that the predicted white is correct, at the possible expense of other color predictions. This means that "perfect" white can be achieved without much degradation in other colors. Informal visual comparisons indicate that this model is the best one to use for computer-generated media.

The masking model was expected to out-perform the linear model whenever there was an issue with channel interaction. However, the model's best performance (on CRT2) is only slightly better than that of the linear model. The primary pitfall of this model is that it depends on constant chromaticity "combined primaries" (CMYK). It is clear from Figure 4 that this assumption is incorrect for the LCD monitors and projectors.

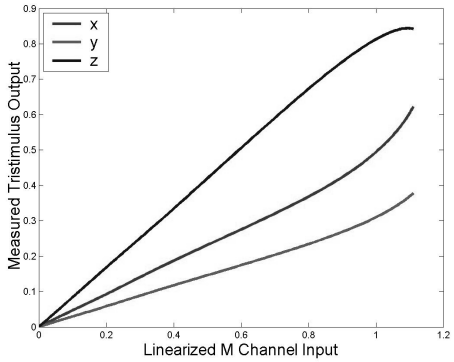

Figure 10. Linearization Failure for the Black Channel on PR1

The chromaticity shift caused by dissimilarity in the shapes of the R, G and B response curves causes the input the linearization step to fail. Figure 10 shows an example of an unsuccessful linearization for the black channel for PR1 in the masking model – note that none of the lines are straight. This explains why the performance of the masking model was better for CRT monitors than any of the other devices – the CRTs do not have the shifting chromaticity problem.

With respect to efficiency, the linear model is the top performer. The linear model is slightly faster than the masking model and nearly 20 times faster than the 3D LUT. The linear model also requires less than half the storage space of the masking model, and less than 1/300th the storage space required for 3D LUT (Table 5).

Table 5. Experimental Running Time and Storage Space as multiples of linear model Usage

	Time	Space
Linear	1.0	1.0
Masking	1.2	2.3
3D LUT	17.0	333.4

Conclusion

Several display characterization models were implemented in this paper: a 3D LUT, a linear model, an extension to the linear model, and a masking model. These characterization models were each tested on seven devices: two CRT Monitors, three LCD monitors and two LCD projectors.

Several general observations were made with respect to collecting characterization data. We found that the phosphor stabilization time on the CRT monitors was much longer than expected, and can take up to 10 seconds. In practice, a delay time of 2500 ms between a colour change and subsequent measurement resulted in acceptable error levels. With respect to integration time, we propose that measurements on CRT monitors be taken with integration times that are multiples of the display scan rate. In addition, it was shown that a training set of 10 points data per axis is sufficient for the linear model (Figure 7).

Although recent papers have indicated that the linear model is not applicable to LCD monitors,[6] it worked well for the LCD devices tested in this experiment. Furthermore, channel interaction was pronounced on the CRT monitors than on several of the LCD displays. The primary issue with the LCD displays was the fact that the response curves for the three input channels were dissimilar, leading to chromaticity shift of combined colours (CMYK). This problem affected the masking model but not the linear model.

Despite these issues, all three models yielded mapping errors of less than 15 ΔE. The 3D LUT model was slightly more accurate than the other models, but it is too cumbersome for actual use. The linear model was the most efficient, with accuracy nearly as good as to the 3D LUT.

The primary drawback of the linear model is that it can be adversely affected by channel interaction. A slight modification to the linear model is presented in the Linear+ model that uses a simple white-point correction technique to ensure correct prediction of white. Our results indicate the Linear+ model is able to guarantee white-point accuracy with minimal degradation for other colours.

References

1. Amidror I. Scattered data interpolation methods for electronic imaging systems: a Survey. J Electronic Imaging2002; 11.2:157-176.
2. Fairchild MD, Wyble DR. Colorime. Colorimetric Characterization of the Apple Studio Display (Flat Panel LCD). Munsell Color Science Laboratory Technical Report, 1998, http://www.cis.rit.edu/mcsl/research/PDFs/LCD.pdf
3. Finlayson G.D., Drew M.S., White-Point Preserving Color Correction, in 5th Color Imaging Conference: Color, Science, Systems and Applications, IS&T/SID, pp.258-261, Nov. 1997
4. Gibson, JE, Fairchild MD. Colorimetric Characterization of Three Computer Displays (LCD and CRT), Munsell Color Science Laboratory Technical Report, 2000, http://www.cis.rit.edu/mcsl/research/PDFs/GibsonFairchild.pdf

5. Kwak, Y, MacDonald LW. Accurate Prediction of Colors on Liquid Crystal Displays. Proc. IS&T/SID 9th Color Imaging Conference 2001; 355-359.
6. Tamura, N, Tsumura N, Miyake Y. masking model for accurate colorimetric characterization of LCD. Proc. IS&T/SID 10th Color Imaging Conference 2002; 312-316.
7. Yasuhiro, Yoshida and Yoichi Yamamoto, Color Calibration of LCDs. Proc. IS&T/SID 10th Color Imaging Conference 2002; 305-311.

Biographies

Bill Cressman received his B.S. degree in Mechanical Engineering from Rice University in 1993, and a Masters in Business Administration from the University of California at Irvine in 2002. He has worked for Hewlett Packard since 1993, in the field of manufacturing data system design. He is currently completing a Masters in Computing Science at Simon Fraser University in Vancouver, BC where his research is focused on calibration and testing of digital displays.

Behnam Bastani received his B.S. degree in Computing Science with a minor in Business from Simon Fraser University in 2003. He has been working in the Intelligent Systems and Data Mining Labs at SFU since 2001, where his work has focused on text clustering and constraint satisfaction. He is currently completing a Masters in Computing Science at SFU, where his research is focused on gamut mapping and characterization of digital colour displays.

Brian Funt is Professor of Computing Science at Simon Fraser University. After receiving his PhD from the University of British Columbia in 1976, he spent 2 years at Stanford as a postdoctoral researcher followed by 2 years as a professor at SUNY Buffalo. He has been at Simon Fraser University since 1980. He is a Marr Prize recipient, and has been Program Co-Chair and General Co-Chair of the IS&T/SID Color Imaging Conference. His work on computational models of color perception began in the early 1980's and has covered color constancy, color object recognition, high dynamic range color imaging, and color in computer vision.