Compensating for non-uniform screens in projection display systems

Siavash Asgari Renani,[a] Masato Tsukada,[b] and Jon Yngve Hardeberg[a]

[a]The Norwegian Color Research Laboratory, Gjøvik University College, P.O. Box 191, N-2802 Gjøvik, Norway
[b]Common Platform Software Research Labs., NEC Corporation, 1753 Shimonumabe,
Nakahara-Ku, Kawasaki, Kanagawa, 211-8666, Japan

ABSTRACT

In this paper the performance of screen compensation based on previous work by Nayar *et al.* and Ashdown *et al.* and five different camera characterization methods are evaluated.

Traditionally, colorimetric characterization of cameras consists of two steps; a linearization and a polynomial regression. In this research, two different methods of linearization as well as the use of polynomial regression up to fourth order have been investigated, based both on the standard deviation and the average of color differences. The experiment consists of applying the different methods 100 times on training sets of 11 different sizes and to measure the color differences. Both CIELAB and CIEXYZ are used for regression space. The use of no linearization and CIELAB is also investigated. The conclusion is that the methods that use linearization as part of the model are more dependent on the size of the training set, while the method that directly convert to CIELAB seems to be more dependent on the order of polynomial used for regression. We also noted that linearization methods resulting in low error in the CIEXYZ color space do not necessarily lead to good results in the CIELAB space. CIELAB space gave overall better result than CIEXYZ; more stabile and better results.

Finally, the camera characterization with the best result was combined into a complete screen compensation algorithm. Using CIELAB as a regression space the compensation achieved results between 50 an 70 percents more similar to the same color projected on a white screen than using CIEXYZ (as measured by a spectrophotometer, comparing absolute color difference in CIELAB) in our experimental setup.

Keywords: projection screen, non-uniformity, projector, camera, ,colorimetric characterization

1. INTRODUCTION

The screen is a vital component of any projection display system when it comes to accurate color reproduction. The ideal projection screen is a perfectly lambertian surface of uniform white, with a reflectivity of one. However, such screens significantly add to the cost of projection systems, and furthermore decrease their versatility and portability. It has been suggested earlier by Nayar *et al.* [1] and Ashdown *et al.* [2] that it should be possible to use a non-ideal screen (including non-uniformities, different textures, etc.), and still achieve good projection quality, by characterizing the non-uniformities by help of a digital camera, and subsequently applying compensating for them by image processing.

In order to set up such a screen compensation system it is necessary to characterize the ways in which both the projection display and the digital camera handle color information.

Nayar *et al.* [1] and Grossberg *et al.* [3] have earlier used a device dependent RGB color space to determine the relationship between the input values fed to the projector and the resulting image captured by a digital camera. Ashdown *et al.* [2] proposed an improvement over the existing methods by taking into account the color gamut of the devices. They used the CIE XYZ color space, as it has the advantage of being device-independent, and their characterization technique involves a 3x4 matrix, found by regression, in a manner similar to the method of Grossberg *et al.* [3].

In the current paper we evaluate the performance of different screen compensation algorithms, with piecewise linear interpolation assuming constant chromaticity (PLCC) model to characterize the projector, and five different methods of camera characterization, as detailed below. The color differences between the compensated images and the original image projected on a white screen are presented. The goals are to explore different camera characterization techniques and to describe the effect on screen compensation in comparable values.

2. CAMERA CHARACTERIZATION

2,1 Methodology

For digital camera devices the process of characterization can be considered to consist of two stages. The first stage performs a linearization, sometimes termed as gamma correction or grey balance, for certain devices. The second stage transforms the linearized values into a device-independent color space, typically CIEXYZ.

For linearization several approaches can been used; we consider here two commonly used methods, the gamma method and polynomial fitting. The gamma method originates from the characterization of CRT monitors; the relationship between device-dependent and independent colors is known to be approximated by:

$$O = k(e - e_d)^{\gamma}, \tag{1}$$

where O stands for the optical signal, e the electric signal, e_d the dark current (also known as the black offset), k a scaling factor and γ the gamma value. This Equation is performed for the three device dependent color channels (i.e. RGB). Since digital camera output is often corrected to yield good images on standard monitors, it is relevant to perform camera linearization based on the gamma method. For cameras the black offset can be obtained by taking a picture with the lens protection cap on and γ can be found by a least-mean square approach [4].

Polynomial fitting is the second method used for linearization. This method uses a polynomial of third order to approximate the data along the curve given by the device dependent and independent data;

$$y = ax^3 + bx^2 + cx + d \tag{2}$$

where y is the device independent data, and x is the dependent data. Just as with the gamma method, it is not certain that this method gives an exact match.

For the conversion from linearized RGB values to a device-independent color space we use 3D polynomial regression of various orders up to fourth order. In addition to performing the conversion directly to CIEXYZ as it is classically done, we also propose to convert directly into CIELAB color space [5], with or without the linearization step. Doing this has the advantage that the regression process minimizes perceptual color difference.

We thus propose the following five methods for further testing:

1. Gamma method used for linearization and regression into CIEXYZ space.
2. Polynomial fitting for linearization and regression into CIEXYZ space.
3. Regression into CIELAB space.
4. Gamma method for linearization and regression into CIELAB space.
5. Polynomial fitting for linearization and regression into CIELAB space.

2.2 Experimental Setup

The equipment used is Canon EOS KISS digital camera, LT35 DLP NEC projector, and Minolta CW-1000 spectrophotometer. Traditionally an imaging target as GretagMacBeth ColorChecker DC or Fuji IT8.7/2 is used for camera characterization. In this case a projector is used. Given that the projector can be spatially non-uniform, the patches were displayed at the centre of projection area. Both the camera and spectrophotometer was aligned with the focus on the centre of the patches. The only light source in the room was the projector. The camera shutter speed (integration time) and aperture values were adjusted so that the camera produced non-saturated values. 33 grey colors were used for linearization and 180 unique and random colors were also chosen, none of them grey. These colors were displayed and measured with the spectroradiometer and re-displayed and captured with the camera. The black offset of the camera was ascertained by capturing an image with the lens cap on.

2.3 Data analysis

Table 1 contains mean ΔE_{ab} prediction error of all methods with increasing order of polynomial regression. The mean ΔE_{ab} in this table is calculated by using all the colors as both training and test set. Our first observation is that the use of 3x3 and 3x5 matrices do not yield acceptable result for all methods. Method 2, 4 and 5 yields acceptable, but visible color difference using a 3x10 matrix. By using 3x20 matrix, method 1 and 2 gives acceptable result as well.

Another observation is the use of the gamma-method and regression into CIEXYZ (method 1) yields worse result than use of polynomial fitting and use of CIEXYZ (method 2) for all matrix sizes. In contradiction with this, method 4 has better result than method 5 when a second order polynomial (3x5) is used. The difference between methods 4 and 1 and between methods 2 and 5 is the use of cubic root function and CIELAB for regression. Lastly, it is observed that method 3, a method with no linearization beyond a cubic root function, outperforms method 1 with a third order polynomial regression and is even better than method 2 when the order is increased to four.

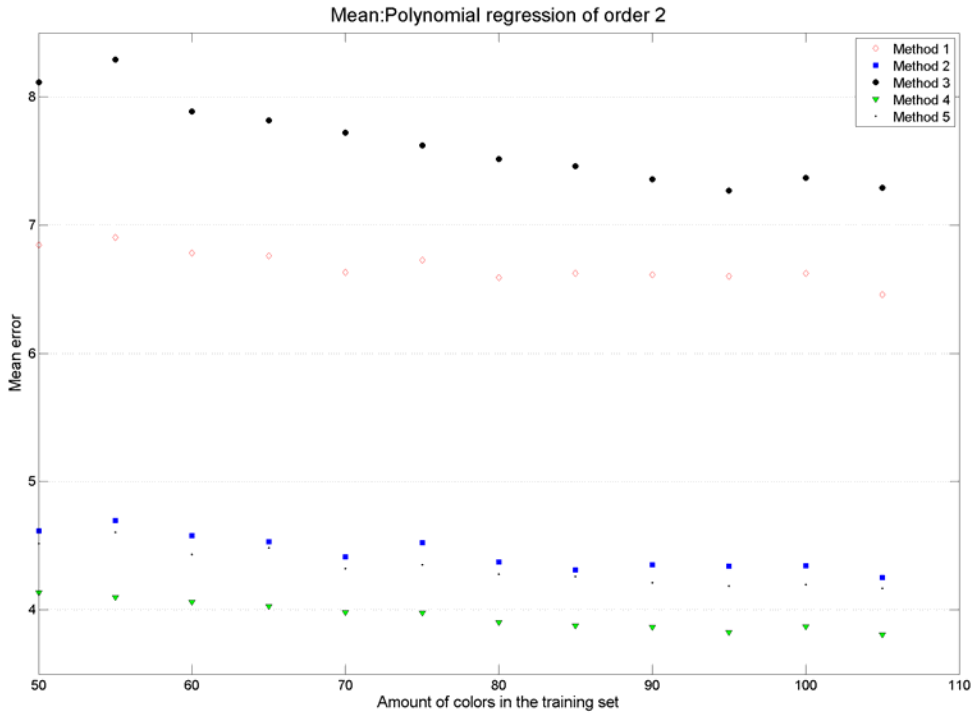


Figure 1: Mean color difference as the training set changes for a second order polynomial regression.

The same trend is observed in Table 2. This table contains the maximum ΔE_{ab} for all the methods using all colors as training and test-set. None of the methods for any order of polynomial gives acceptable difference, except for Method 4 and Method 5. Fourth order polynomial gave lower maximum for all methods. Method 2 can directly be compared to former research [6]. From this comparison it can be seen that the results obtained deviates a little; mean and max color difference with use of fourth order regression is lower than by using third order regression. Another deviation is lower mean and max error when a third or fourth order regression is used. Considering both the mean and maximum error, Method 4 performs best but Method 5 is almost equally good. 3x3 and 3x5 matrixes had unacceptable color difference so they are not discussed further. For the rest of the chapter the focus is on second, third and fourth order regression.

Figure 1, 2 and 3 displays the result of mean ΔE_{ab} of the five different methods using different training sizes. The methods are applied 100 times per training size, with different training-set each time. The reported error is the average ΔE_{ab} of each application of the methods.

Table 1: Mean ΔE_{ab} using all colors for test and training.

Size of regression matrix	Method 1	Method 2	Method 3	Method 4	Method 5
3x3	10.35	7.77	19.66	9.03	7.80
3x 5	8.11	7.18	16.29	8.21	6.18
3x10	6.20	3.97	6.58	3.51	3.75
3x20	4.52	2.24	2.82	1.79	2.54
3x35	3.20	1.40	1.34	1.10	1.38

Table 2: Max ΔE_{ab} using all colors for test and training.

Size of regression matrix	Method 1	Method 2	Method 3	Method 4	Method 5
3x3	72.94	34.68	85.42	39.32	39.16
3x 5	28.82	26.59	80.23	33.64	35.77
3x10	31.87	17.87	31.95	15.57	25.59
3x20	28.78	11.98	17.45	7.96	13.46
3x35	22.25	10.07	10.41	4.26	4.32

Figure 1 presents the result for the use of a second order matrix, Figure 2 for a third order matrix and Figure 3 for a fourth order matrix. It can be observed that these figures mostly support earlier observations made from Table 1 and Table 2. Method 2, 4 and 5 are at an acceptable level, while method 1 and 3 have higher mean ΔE_{ab} for a second order polynomial for all training-sets. For the third order polynomial there is a contradiction from former observations; the mean error for method 5 does not fall below three and have higher average error than method 2.

Otherwise as the number of patches in the training set increases, the average errors comply with what is already seen from Table 1. The figure for fourth order polynomial also mostly supports earlier observations; as the amount of colors increases the mean error for the methods using linearized values and CIELAB for regression and Method 2 drop below three. The largest irregularity from previous observations can be seen for Method 3.

For some of the methods, the error becomes close to the error attained by using all colors as training-set, but the figures also suggest that this can be further improved by increasing training-set size. A size increase had the largest effect for a fourth order regression as the error drops exponentially in the beginning.

Further focus is on error distribution of two of the methods; 2 and 4. Method 2 performed best of the methods using CIEXYZ and method 4 did the same for CIELAB. The same training set is used for both methods. Considering that a lesser order polynomial is more computationally efficient and a third order regression gave sufficient result, it was decided to also focus only on third order regression. Figure 4 shows a histogram over color difference (ΔE_{ab}) to the left and to the right original and predicted color values is plotted in an *a*b** diagram. From the histogram it can be observed that large number of colors have errors which are potentially not perceivable ($\Delta E_{ab}<3$). The actual amount is 73.3 percent of the test-set. 87.6 percent of the colors are considered to have an error of acceptable or better ($\Delta E < 6$). The error is mostly located between $b^* = [35, 90]$ and $a^* = [18, 45]$. The mean error for this particular transformation is 2.4 and 100 colors were used to calculate the transformation matrix. Same type of Figure can be found for method 4, namely Figure 5. Here 76.2 percent is considered to be non-perceivable and 92.4 percent is considered to acceptable. The error is more spread in *a*b** space but it can be located mostly for $b^*>40$ and $b^*<-20$.

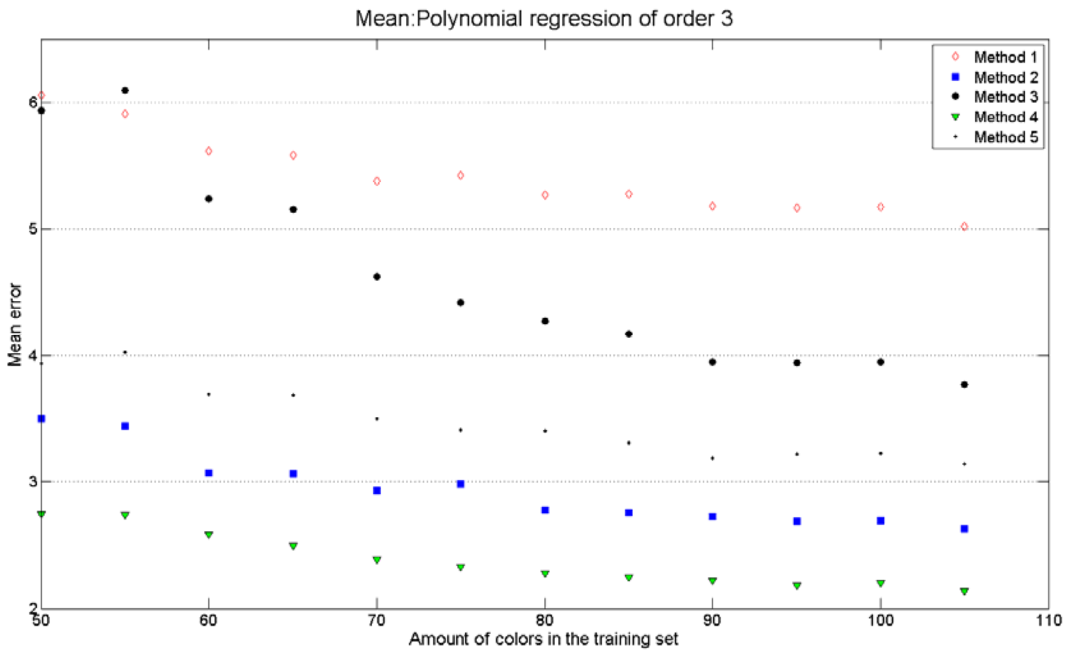


Figure 2: Mean color difference as size of the training set increases for a third order polynomial regression.

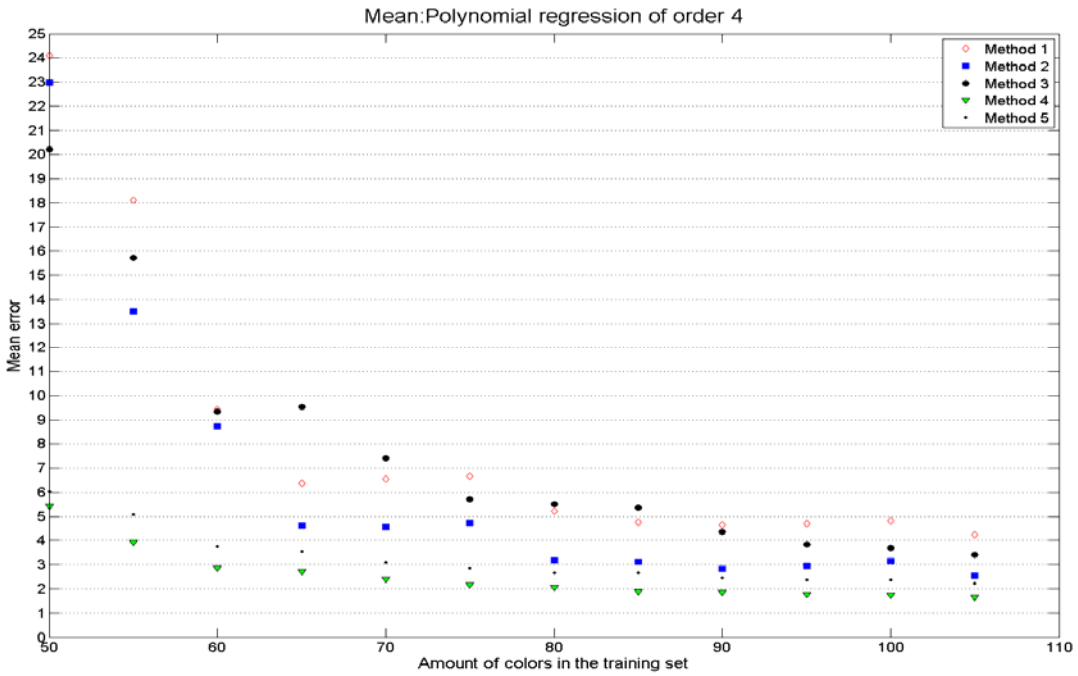


Figure 3: Mean color difference as size of the training set increases for a fourth order polynomial regression.

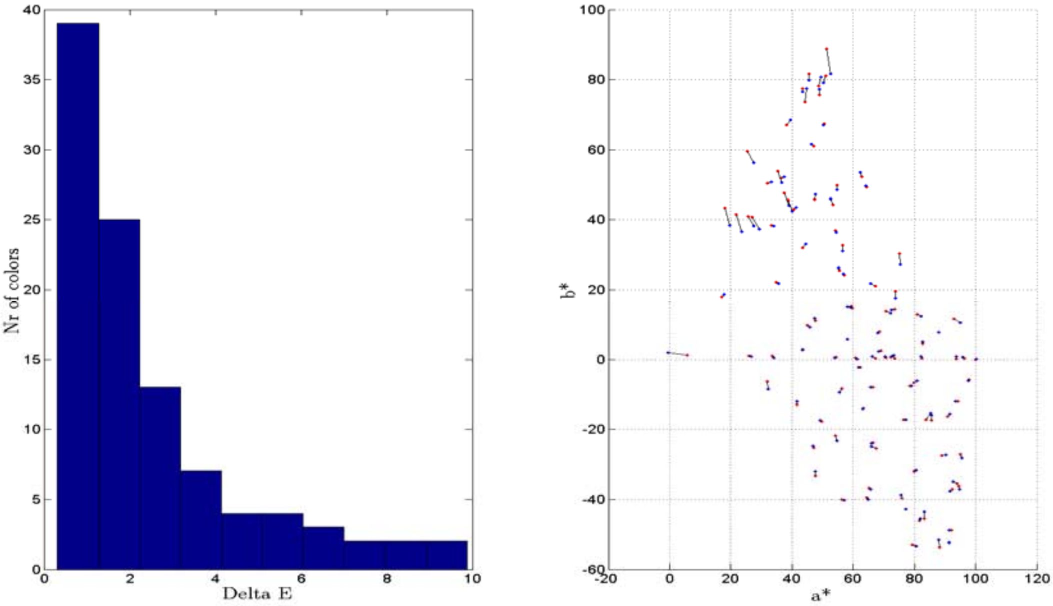


Figure 4: Error distribution of method 2 using a third order polynomial. To left is a histogram over ΔE_{ab} and right is plot of predicted and original *a** and *b**; Red is calculated from original measured CIE XYZ and blue is the predicted.

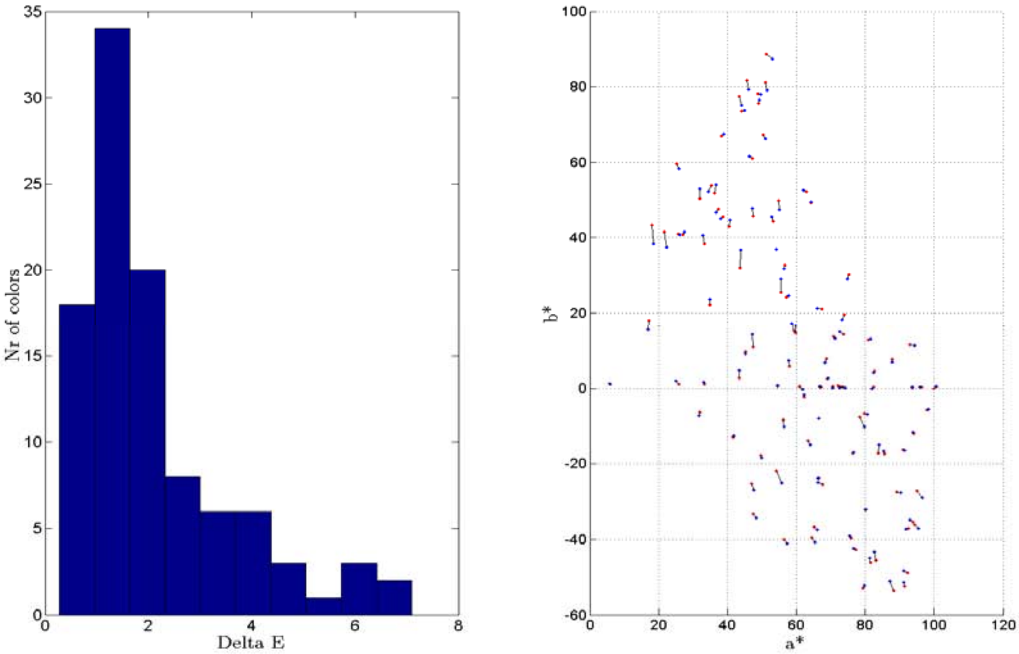


Figure 5: Error distribution of method 4 using a third order polynomial. To left is a histogram over ΔE_{ab} and to right is plot of predicted and original *a** and *b**; Red is calculated from original measured CIE XYZ and blue is the predicted.

2.4 Standard deviation of error

The standard deviation of the error for the second order polynomial regression (Figure 6) shows a lower deviation for methods which performed well, which is to be expected. The maximum variance could be found for method 3, a method that is not linearized beyond the cubic root function. No similarities can be seen between methods that use the same approach for linearization.

For the third order polynomial regression it is expected a greater variance than for second order due to an increased degree of freedom in the equations. Figure 7 shows that this is not always true; the standard deviation for method 4 and 5 are almost equal. The fourth order regression supports the assumption that number of equations increases the standard deviation (Figure 8). It can be clearly seen that if less than 85 colors are used for the training-set the standard deviation becomes significant. For method 3 the standard deviation remains high for all the training-sets but it is also observed that other methods have almost equal variance at some points.

For all orders of polynomial regression the methods 4 and 5 are the most stable; never reaching three and mostly below one. If one considers that the standard deviation should not exceed 25 percent of the threshold for "hardly visible"– 0.5 – only three methods becomes eligible for second and third order regression; method 2, 4 and 5. For fourth order, only method 4 and 5 have low enough variance.

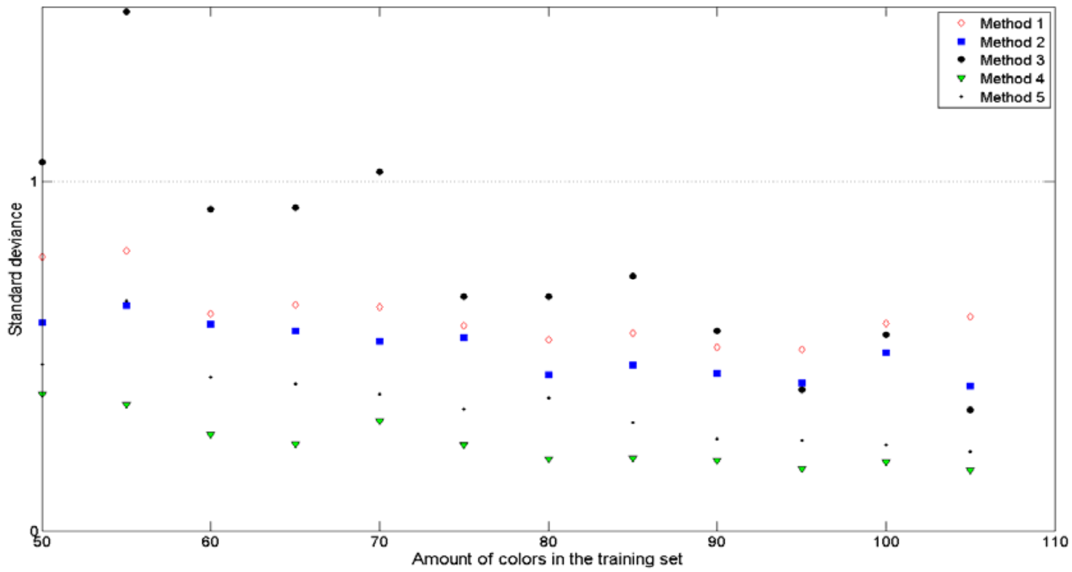


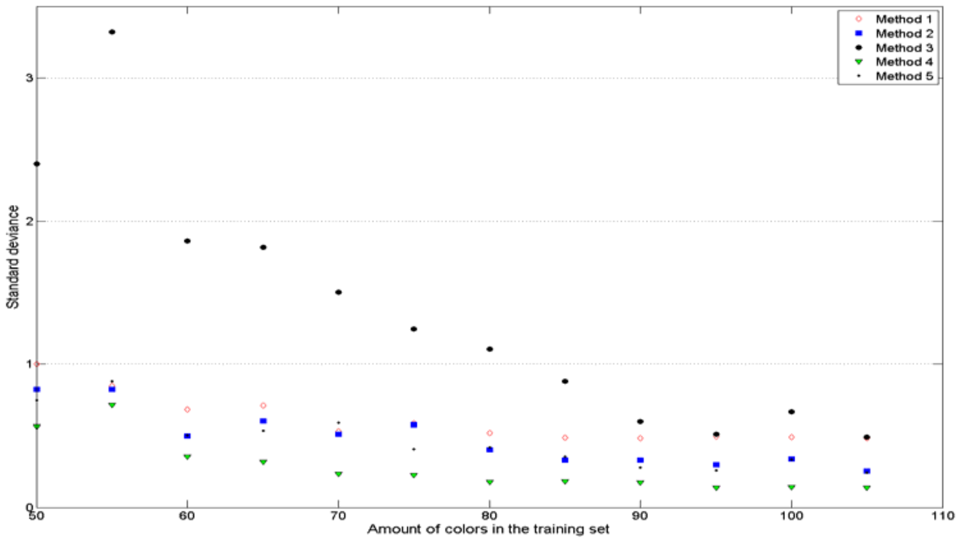
Figure 6: Standard deviation as the training-set changes for a second order polynomial regression.

2.5 Conclusion on camera characterization

From the analysis of the results it is observed that methods which use CIELAB for regression gave best result when all colors were used in the training-set and test-set. Both average and maximum color difference were lower. When variable amount of colors are used for the training set, the difference between using CIELAB and CIEXYZ color spaces becomes less pronounced. Of those methods with the same linearization approach, Method 4 gave far better result than method 1, but method 2 in some cases gave better or equal result to method 5. These differences were however small, and the overall conclusion is that CIELAB performed best.

It can also be observed that methods with same linearization technique but using the two different color spaces performed differently. Method 2 performed better than method 1, yet method 4 performed better than method 5. This leads to the conclusion that linearization techniques that performs well in CIEXYZ does not necessarily perform equally well in CIELAB.

The importance of linearization as part of the characterization method is stressed by the result of method 3. Using the cubic root function on the device-dependent values directly has potential to give a good result, and although it might be tempting to avoid the arduous task of linearization, the variance of the result should be considered. It was observed that Method 3 was the most unstable method. As for all methods, the stability increased as amount of colors in the training-set increased but for Method 3 the increase was more drastic. This suggests that larger training-set is needed when no linearization techniques are used



.

Figure 7: Standard deviation as the training-set changes for a third polynomial regression

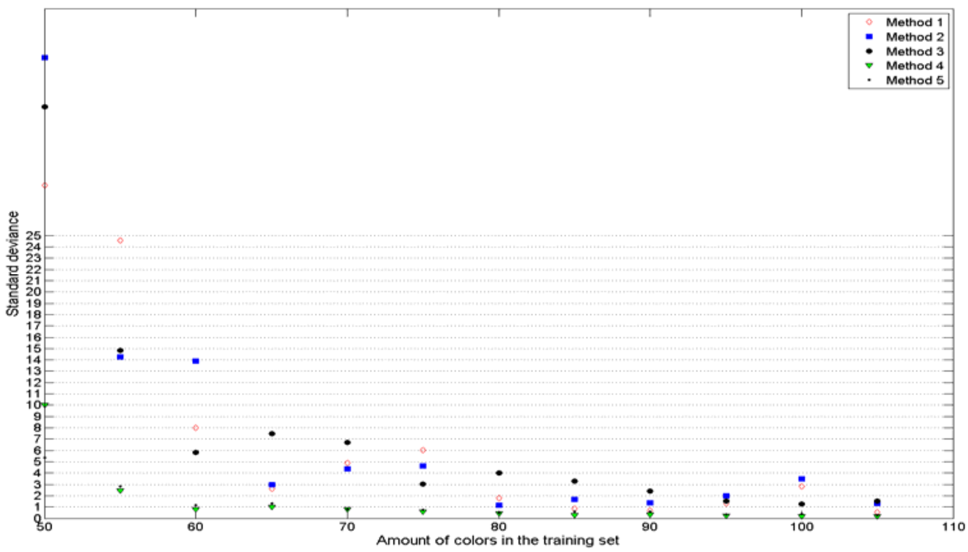


Figure 8: Standard deviation as the training-set changes for a fourth order polynomial regression.

3. SCREEN CHARACTERIZATION AND COMPENSATION

Figure 9 shows a simplified version of the model proposed by Nayar *et al.* [1]. Conceptually, the projected colors are first modified by the projector and its color characteristics. Depending on which position a particular color is projected to, the color becomes further modified by the screen, before it is finally captured by the camera. It is assumed that the modification induced by the screen can be described as

$$c = A_s p, \tag{3}$$

where p is projector output, c is camera input, and A_s is a 3x4 matrix, which can be calculated by a least-square regression approach. Since the screen can be non-uniform in color, different transformation matrices must be derived for each pixel. From the camera characterization it has been observed that regression in CIELAB gave good results. To gain the same advantages for screen characterization this space was also used for regression here.

The method for finding the transformation matrix for the screen using regression can be described in six steps:

- Project images on the screen.
- Capture them with the camera.
- Transform pixel coordinates to original coordinates using a geometrical alignment method [1].
- Transform colors from camera-dependent values to CIELAB.
- Transform colors from projector-dependent values to CIELAB.
- Least square regression to obtain transformation function for screen modifications per pixel.

For compensating for the screen, the inverse modification has to be applied to the image which is intended to be projected. Thus, it can be more desirable to derive directly the inverse transformation function B_s in the equation

$$p=B_s c, \tag{4}$$

since it finds the inverse transformation directly and there is no need for using for instance a pseudo-inverse. The screen compensation can then be described by Eq (6) where the inverse transformation function of the screen, B_s, is applied to projector output; p. The projector output is then inverted to projector input (projector independent values) after compensation using the inverse transformation function obtained by characterization.

$$O_{comp} = B_s p \tag{5}$$

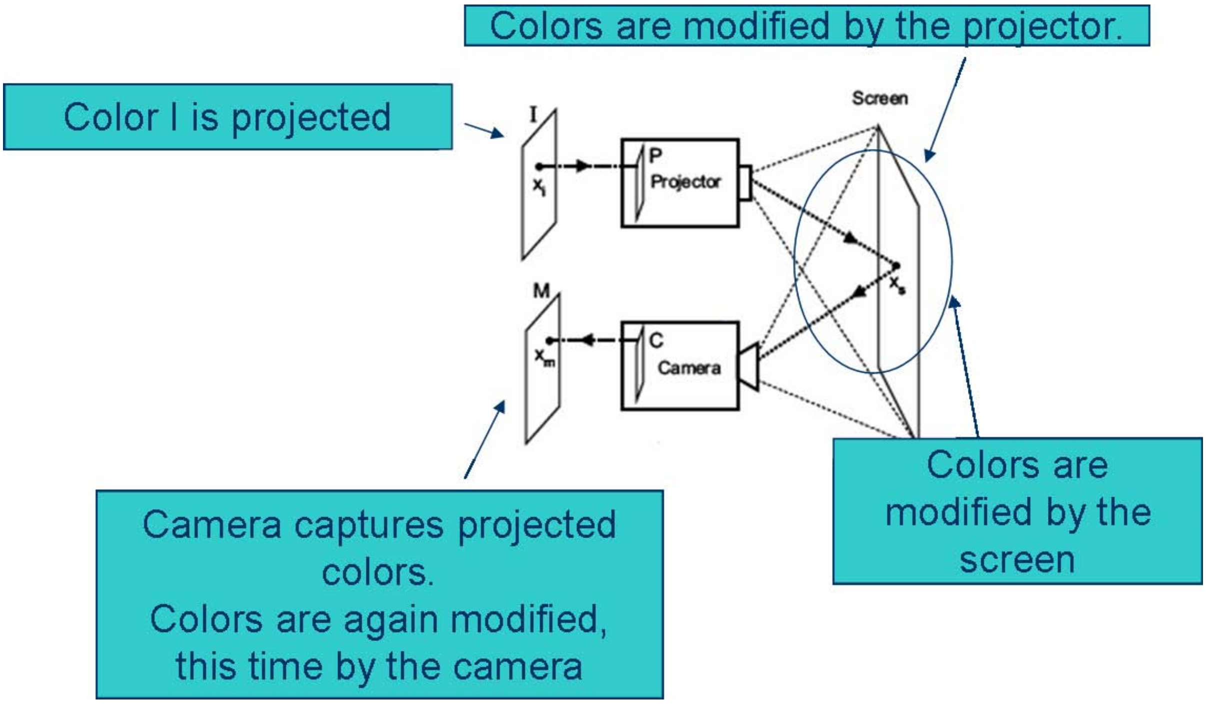


Figure 9: Overview of color modifications in a projector-camera system.

Neither the projector characterization nor the camera characterization compensates for non-uniformity caused by these devices. A_s should, theoretically, compensate for these as well. It should be mentioned that since non-uniformity caused by the camera can only be perceived on captured images, any compensation that includes this can affect a compensated image negatively if it is perceived directly on the screen. It would include compensation for something that cannot be seen with the eye alone.

3.1 Experiment

For camera characterization the transformation using Method 4 were used. This method directly transforms device-dependent colors to CIELAB space. The transformation function is derived using 105 colors and a third order polynomial. The mean error is 2.2 ΔE_{ab} units. The PLCC display characterization model is used to derive the transformation function for the projector. 20 colors are used for linearization and Spline interpolation. The obtained mean error is 1.2 ΔE_{ab} units. Bear in mind that both characterization for the camera and projector are done in the same condition. The same conditions are also used for the screen compensation.

A non-uniform screen was created and printed using a large format inkjet printer, a HP DesignJet 1050c. The screen contained three parts; Blue, white and green. The blue had CIELAB values of [54.4, 11.86, -25.01], and the green part had CIELAB values [69.84 -36.10 28.32]. The colors were visibly saturated but not overly saturated. These colors were chosen on the assumption that they are contained in the gamut of projector and camera. Figure 10 is a picture of the screen, geometrically aligned. The edges are “jarred” because of the geometrical alignment method used.

Nayar *et al.* [1] suggested projecting two colors per channel onto the screen but also advised more colors for increased precision. It was decided to use 5 colors per channel. As the number of colors per channel is increased, more calculation is needed and rapidly increases the time needed for calibration to the point of impracticality.

To evaluate the compensation, it was decided to project random colors, and calculate error for the three parts of the screen independently. The camera was used to capture images of both compensated and uncompensated projections. A total of 20 uniformly colored images were chosen as part of the test-set. Each image contained around 490000 pixels. The first ten images were chosen randomly. The remaining ten were 50 percent more saturated version of the previous ten. It was further decided to calculate ΔE_{ab} error for each part; white, blue, white and green, separately. The error calculation is done between original and compensated and original and uncompensated. The captured images are geometrically aligned to the screen and transformed to CIELAB values using the transformation function used for the camera. The original image is transformed to CIELAB values using the transformation function found for the projector.

Figure 10: The screen. Geometrically aligned so that the surface used for projecting colors is shown.

Table 3 displays the percentage of improvement the compensated image has over the uncompensated for the first ten random uniform colored images. It can be observed that for the white in the middle improvement exists. This is considered to be because of the white of the paper is different than the one used for the characterization. It can also be speculated that the radiation from blue and green parts interferes with the radiation from the white part. The mean percentwise improvements for the three parts are 70 percent for the blue, 58 for the white and 49 for the green.

Maximum errors are improved in average 61.4 percent for blue, 51.2 for white and 44.6 for green. For one of the colors the compensation has created a larger difference rather than a smaller one. The percentwise improvement is displayed in Table 4.

For more saturated colors percentage improvements are displayed in Table 5. The mean percentwise improvement is 65 percent for the blue, 52.3 for the white and 49 for the green. Again, in the white area the compensation failed for one color. Maximum color differences for the saturated images are displayed in Table 6. The mean percentage (Table 5) improvement is 52.4 for the blue, 38.02 for the white and 42.6 for the green.

Table 3: Percentwise improvement for the compensated over the uncompensated for mean error. Ten random uniform images.

Uniform images	Blue	White (middle)	Green
Color 1	93.6	53.8	59.6
Color 2	89.9	77.9	49.0
Color 3	56.2	6.6	76.0
Color 4	85.9	22.2	71.7
Color 5	73.3	77.6	34.5
Color 6	78.9	19.3	70.0
Color 7	70.9	71.1	8.3
Color 8	84.3	89.3	48.7
Color 9	33.0	88.5	38.0
Color 10	29.4	78.2	33.9

Table 4: Percentwise improvement for the compensated over the uncompensated for max error. Ten random uniform images.

Uniform images	Blue	White (middle)	Green
Color 1	82.3	61.0	47.3
Color 2	31.3	55.0	37.0
Color 3	54.8	45.4	64.8
Color 4	76.0	33.7	36.3
Color 5	21.4	-7.6	36.3
Color 6	81.3	30.4	73.1
Color 7	9.9	69.5	45.7
Color 8	91.4	83.6	44.4
Color 9	90.7	86.7	31.4
Color 10	52.7	65.3	44.7

Table 5: Percentwise improvement for the compensated over the uncompensated. For 50 percent more saturated colors.

Uniform images	Blue	White (middle)	Green
Color 1	83.7	45.9	52.6
Color 2	82.6	67.9	45.5
Color 3	53.4	-7.1	74.8
Color 4	79.4	13.7	66.2
Color 5	58.2	73.2	42.9
Color 6	69.2	12.3	61.7
Color 7	67.5	65.8	6.6
Color 8	69.4	82.6	41.0
Color 9	28.4	84.8	27.0
Color 10	22.3	73.2	28.2

Table 5: Percentwise improvement of max. color difference for the compensated over the uncompensated. For 50 percent more saturated uniform colored images.

Uniform images	Blue	White(middle)	Green
Color 1	54.8	45.2	48.6
Color 2	33.2	47.6	31.7
Color 3	33.3	14.8	52.6
Color 4	66.1	22.4	36.3
Color 5	-6.3	-39.9	36.3
Color 6	68.0	20.5	73.1
Color 7	55.4	61.2	45.7
Color 8	82.2	71.4	44.4
Color 9	86.1	79.8	31.4
Color 10	51.4	57.1	44.7

4. CONCLUSION

In this paper we have proposed and evaluated several ways to compensate for a non-uniform screen in a projection display system, with the use of a digital color camera as a measurement instrument.

Based on the experimental data we see that 50-70 percent improved non-uniformity in terms of average color differences is achieved using regression in CIELAB space. The improvement on maximum color difference is smaller; between 38 and 53 percent.

Although this improvement is considerable, a perceivable difference is still evident when comparing the original image projected on a white screen and the compensated image on a non-uniform screen. The mean difference (in ΔE_{ab}) is 12.05 for blue, 12.34 for white and 11.01 for green. For an uncompensated image the mean difference is 36.64 for blue, 30.91 for white and 23.69 for the green part. The compensated image makes the image more similar to the original image compared to the uncompensated but the screens effect is still visible.

REFERENCES

[1] Nayar, S.K., Peri, H., Grossberg, M.D., and Belhumeur, P.N., "A Projection System with Radiometric Compensation for Screen Imperfections," in [ICCV Workshop on Projector-Camera Systems (PROCAMS)], (2003).

[2] Ashdown, M., Okabe, T., Sato, I., and Sato, Y., "Robust content-dependent photometric projector compensation," in [Proc. 2006 Conference on Computer Vision and Pattern Recognition Workshop], 6 (2006).

[3] Grossberg, M.D., Peri, H., Nayar, S.K., and Belhumeur, P.N., "Making one object look like another: Controlling appearance using a projector-camera system," in [IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)] Volume 1, 452-459 (2004).

[4] Hardeberg, J.Y., Seime, L., and Skogstad, T., "Colorimetric characterization of projection displays using a digital colorimetric camera," in [SPIE proceedings 5002, Projection Displays IX], 51-61, (2003).

[5] Hardeberg, J.Y., "Acquisition and reproduction of colour images: colorimetric and multispectral approaches", PhD thesis, Ecole Nationale Superieure des Telecommunications (1999).

[6] Chung, V., Westland, S., Connah, D., and Ripamonti, C., "A comparative study of the characterization of colour cameras by means of neural networks and polynomial transforms," Coloration Technology 120, 19-25 (2004).