

Evaluation of Color Correction Methods for Printed Surfaces

Maliha Ashraf

Univeristé Jean Monnet

Océ Print Logic Technologies S.A.

Créteil, France

maliha.ashraf@etu.univ-st-etienne.fr

Luis Ricardo Sapaico

Océ Print Logic Technologies S.A.

Créteil, France

ricardo.sapaico@oce.com

Abstract—Achieving truthful reproduction of color is a well-known problem for printing applications. In this paper, we evaluate different methods for camera color calibration, within the scope of a moving image acquisition system. The acquisition device consists of a RGB camera and two fixed tungsten lamps. Our goal is to determine the most robust end-to-end framework given the acquisition conditions. We tested some color mapping methods commonly mentioned in the literature and made modifications to our specific needs. Our modified 3D lookup table (LUT) based method performs the best in the given conditions with future possibilities of hybrid LUTs combining the best performing aspects of different types of LUTs.

Index Terms—color calibration, color correction, color imaging, camera calibration

I. INTRODUCTION

For print quality inspection, obtaining accurate color is essential. However the acquired camera responses are not a direct depiction of the objects reflectance values. In a typical RGB camera it is a function of the source illuminance (including effects such as specular reflection), objects surface reflectance, and the camera response function [1]. Apart from the environmental factors, camera parameters such as exposure settings (aperture, shutter speed, etc.) also influence the camera response making it non-linear with respect to input radiometric values. Additional non-linearities are introduced during the camera post-processing step. Therefore, in our experiments we will use the raw images to keep the response values proportional to the recorded intensities.

Moreover, we need the color measured from the camera to represent the human visual system response. But since the real world camera spectral sensitivity functions are not linear transformations of human cone spectral sensitivities (i.e. do not satisfy Luther conditions) [2], it is not possible to directly extract the equivalent color as seen by the human eye from the camera's response.

Color calibration is essentially the process of finding the transformation function that converts the device-dependent responses to a standard device-independent color space. In our case, we require it to be the CIE LAB color space. The goal is to find a mapping function, that will convert the camera RGB response (under fixed conditions) to its corresponding L*a*b* triplet. The mapping function could

either be a direct conversion between device RGB and L*a*b or it could use an intermediate color space such as XYZ to relate the two color spaces. Color correction is the process of applying the aforementioned transformation function to the camera response to convert it into the equivalent target color space.

Some additional constraints specific to our application are the small working distance, and non-uniform illumination. The goal of the application is to create an accurate representation of a flat object. We will develop a complete pipeline from acquisition to color correction and provide an overview of different color mapping algorithms modified to fit our application. In Section II, we will discuss some of the previously done researches on color correction methods. In Section III, the overview of our equipment and the description of pre-processing, color calibration and correction steps is given. Sections IV and V present and summarize the findings and give recommendations for future research.

II. RELATED WORK

Extensive research has been aimed at solving the problem of truthful color acquisition by cameras. Some of the most relevant studies are discussed here.

Least square approximation (LSA) methods aim to minimize the residual error between the ground truth colorimetric values and the predicted colorimetric values when transformed with a transformation matrix. Depending on the desired mapping inputs and outputs, and the mapping function, there can be several variations of LSA methods. Some of the commonly used LSA methods are linear, polynomial, and root-polynomial approximations.

Linear mapping is one of the earliest used, and the simplest LSA algorithm [3]. As depicted in Fig. 1, the relationship between RGB camera response and the XYZ device-independent is roughly linear (see R-X, G-Y, and B-Z plots) and can be represented mathematically as:

$$Y = XC^t \quad (1)$$

where:

Y = Matrix of device-independent colorimetric values

X = Matrix of any function of camera response

C = Transformation matrix; coefficients of linear solution

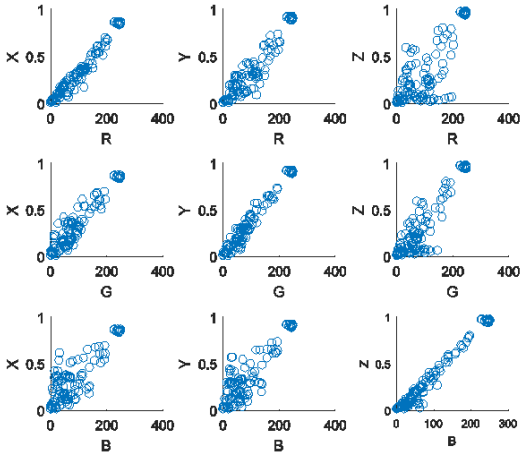


Fig. 1: Relationship between corresponding RGB and XYZ tristimulus values

A 3x3 transformation matrix between the RGB and XYZ values can be calculated by least square approximation or minimizing the residual norm of Eq. 1. The closed-form solution of the following relationship can be obtained by Moore-Penrose pseudo-inverse method:

$$\min_c \|XC^t - Y\|^2 \quad (2)$$

Because the linear camera responses are directly proportional to the scene radiance/device exposure, the linearly mapped transformation is also robust to irradiance/exposure changes. Linear mapping although simple is not always a correct representation of the relationship between the two corresponding especially in low chromaticity regions. To obtain a more suitable mapping, the transformation can be modelled as a higher degree polynomial between XYZ tristimulus values and function of camera RGB values. This takes into account the inter-channel relationships as well. If the exposure and environmental conditions are kept constant, polynomial regression works fairly well [4]. However, care must be taken when using higher-degree polynomial models as it could result in over-fitting. Moreover, the transformation is no longer exposure invariant as the matrix X in Eq. 1 is no longer a direct representation of the irradiance/exposure.

In order to preserve the exposure invariance property and map a higher degree transformation, root polynomial terms are added to the simple linear matrix X in Eq. 1. Examples of second order root polynomial terms are \sqrt{RG} , \sqrt{GB} , and \sqrt{BR} . Since the degree of the added polynomial terms is kept as 1, they are still proportional to changes in irradiance or exposure [5].

Extended Linear Color Correction (ExLCC) approach used by Finlayson and Johnson [6] exploits the fact that the CIELAB coordinates are converted from CIEXYZ independently from each other. For example the lightness L^* is only dependent on Y , a^* is dependent on X and Y , and b^* is dependent on Y and Z . Hence, instead of aiming to minimize the overall color difference E_{00} , it is better to individually

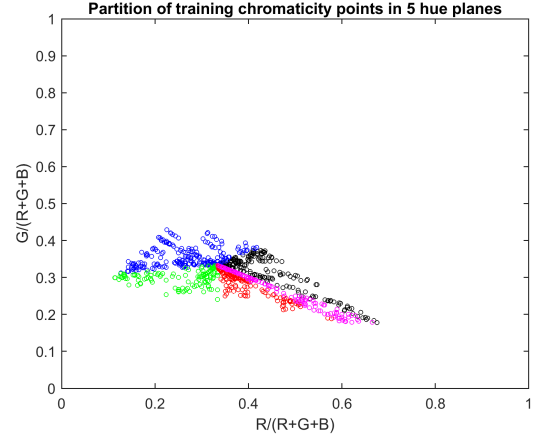


Fig. 2: Relationship between corresponding RGB and XYZ tristimulus values

minimize the difference in each parameter. Thus, instead of calculating one generalized transformation matrix, this method calculates three transformation matrices which independently calculate L^* , a^* and b^* values.

Hue-plane Preserving Color Correction (HPCC) takes into account the non-uniformity of perceptual color spaces. The color sensitivity of the human visual system is different for different parts of the gamut [7] and so it can be more useful to calculate different mapping transforms dependent on the position of the color in its corresponding gamut. Instead of mapping the absolute camera responses, the chromaticities or weighted RGB values are used to segment the color spaces in areas of similar hues as shown in Fig. 2. Separate transformations for each hue plane are calculated with additional optional constraints of white-point and boundary color constancy [8].

Multidimensional Lookup Table (MLUT) based methods also exploit the non-linearities of the color spaces. Instead of a single generalized transformation matrix for the whole gamut, it is divided into a grid of representative color coordinates. Measurements are made for a sparse set of selected grid points and the rest of the points are estimated using non-linear interpolation techniques [9].

III. EXPERIMENT

A. Equipment

a) *Camera*: The camera is mounted perpendicular to the object table at a height of 16.1 cm. We opted for the UI-1490LE-C-HQ, a lightweight 10.55 megapixels USB 2.0 camera by iDS. It has a CMOS sensor from ON Semiconductor with a resolution of 3840 x 2748 pixels and a rolling shutter. The lens has a focal length of 3.5 mm and f-stop range from F2.4~14. With such small working distance, distortions such as vignetting, and uneven spatial response of the sensor are also more pronounced. Hence, they need to be compensated in the pre-processing steps.

b) *Illumination*: Two tungsten lamps are placed on either side of the object table, positioned to minimize the shadows and provide an approximately homogeneous illumination.

Flat-field correction is applied to compensate the remaining illuminant non-uniformity. Another problem is the specular reflection due to change in illuminant geometry across the object table. To compensate, the polarizer's angle is adjusted to minimize the specular reflection at different positions on the object table.

c) *Calibration Target*: Since our application is geared towards printed surfaces, an IT8/7.3 CMYK calibration target is used to create the calibration dataset. The printable image of the calibration target is shown in Fig. 3. The target has 928 color patches covering the entire Pointer's gamut [10].

d) *Spectrophotometer*: Spectral measurements are taken using the X-Rite i1Pro spectrophotometer. The $L^*a^*b^*$ values are computed for a 2° observer under D65 illuminant.

B. Image Processing Pipeline

All the image processing and color calibration/correction steps are performed using MATLAB. In this section, the image processing steps applied before color characterization are described briefly. These steps are common for calibration target images as well as any image which is to be color corrected. The flowchart of the processing pipeline is depicted in Fig. 4.

1) *Demosaicing*: Since the captured images are 8-bit raw images, demosaicing must be performed to convert them into 24-bit RGB images. The default MATLAB `demosaic` function is based on linear interpolation and produces visible artefacts, especially in large areas of uniform color, such as the flat-field images. Instead color plane interpolation using alternating projections method [11] was used which albeit slower produced no visible artefacts.

2) *Flat-field correction*: Flat-field correction is necessary to correct inter-frame (illuminant differences between images acquired at different positions on the object table) as well as intra-frame (illuminant differences between different areas of the same image) non-uniformity. Thermal noise and dark current noise is also removed during this process. The pixel-wise operation of flat-field correction can be represented using Eq. 3:

$$C(x, y) = \frac{R(x, y) - D(x, y)}{F(x, y) - D(x, y)} \quad (3)$$

where:

- (x, y) = Position of an arbitrary pixel
- C = Corrected image
- R = Demosaiced image
- D = Dark image
- F = Flat-field image



Fig. 3: Calibration target

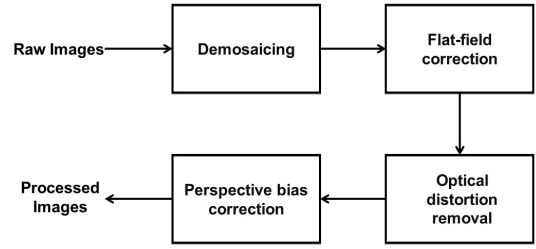


Fig. 4: Image processing pipeline

3) *Optical distortion removal*: The camera's intrinsic parameters (focal length, principal point, skew, radial, and tangential distortion coefficients) are estimated using Bouguet's toolbox [12]. The estimated distortion model is then applied on the images to remove optical distortions.

4) *Perspective bias correction*: In practical settings, it is not possible to position the camera completely parallel to the surface normal. The effect of perspective bias is especially pronounced when the image acquisition distance is small. A checkerboard reference pattern is used to calculate homography (transformation function) that registers the image captured by the camera to a plane perfectly parallel to the object table.

C. Color Calibration and Correction

The image of calibration target is first processed through the pipeline described in Section III-B. The RGB values of all 928 patches along with the corresponding spectrally measured values are extracted to create the calibration dataset. The convex hull of the 928 data points in RGB space is shown in Fig. 5.

The color mapping algorithms described in Section II were tested to find the most suitable one for our application. In addition to the methods existing in previous literature, another novel mapping algorithm based on direct relationship between RGB and CIELAB color space was tested. Instead of mapping RGB values to XYZ then converting the values in their corresponding CIELAB values, we can directly map a relationship

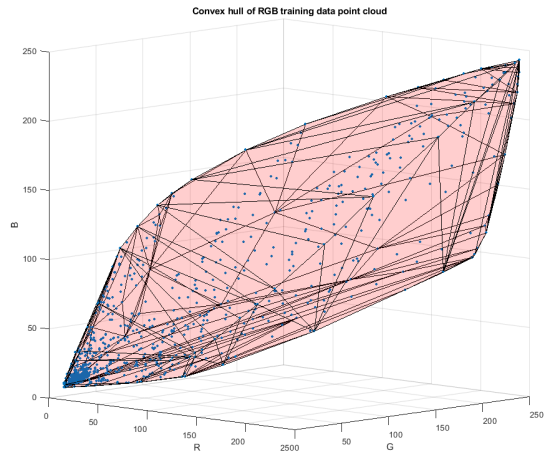


Fig. 5: Convex hull of calibration dataset

between RGB and Lab. The transformation between XYZ and CIELAB can be represented by the following equation:

$$\begin{aligned} L^* &= 116(Y/Y_n)^{1/3} - 16 \\ a^* &= 500\{(X/X_n)^{1/3} - (Y/Y_n)^{1/3}\} \\ b^* &= 200\{(Y/Y_n)^{1/3} - (Z/Z_n)^{1/3}\} \end{aligned} \quad (4)$$

where:

$$\begin{aligned} [L^*, a^*, b^*] &= \text{Transformed coordinates in CIELAB} \\ &\quad \text{color space} \\ [X, Y, Z] &= \text{Input coordinates in CIEXYZ color space} \\ [X_n, Y_n, Z_n] &= \text{Coordinates of white point in CIEXYZ} \\ &\quad \text{color space} \end{aligned}$$

It can be noted from Eq. 4 that the relationship between the two CIE color spaces is a cube-root function [1]. We have previously established in Section II that XYZ and RGB color spaces are approximately linearly related (Fig. 1). If we roughly consider RGB color space analogous to XYZ color space, it can be deduced that the transformation between RGB and CIELAB can also be a cube-root function. To estimate the mapping via LSA method, we first transformed the RGB values to their corresponding functions of cube-root and then solved the closed-form least square equation. For cube-root polynomial LSA, the parameters of matrix X in Eq. 1 are $(\sqrt[3]{R}, \sqrt[3]{G}, \sqrt[3]{B})$, while matrix Y has the corresponding $L^*a^*b^*$ values from the calibration dataset.

For the HPC method, testing was done with a different number of hue plane segments. The best results were obtained when the data points were divided in 9 segments. Further increasing the number of unique hues resulted in decreased accuracy in color correction due to over-fitting.

Two lookup tables (LUTs), one based on absolute RGB to XYZ transformation (LUT 3D), and the other based on relative rg chromaticity to XYZ transformation (LUT 2.5D), were generated. The LUT grid was initialized for each possible RGB value $[0, 255]$ (16.78 million) combinations. However, since our gamut is bounded by the colors of real-world surfaces, it

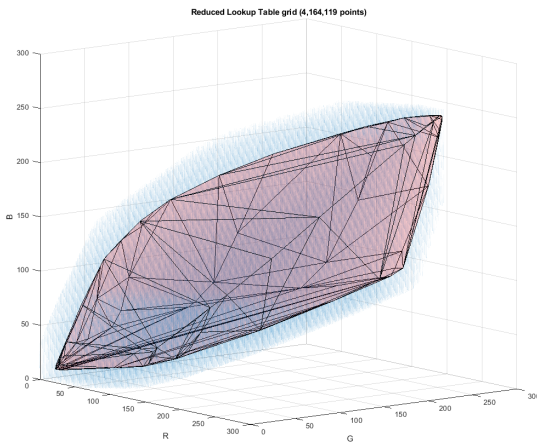


Fig. 6: Expanded gamut for LUT (blue dots)

can be reduced. To reduce the gamut, we expanded our training gamut (Fig. 5) to include the points which are within 20 units (RGB space) distance from the convex hull. The expanded grid for the LUTs consisted of ~ 4 million points (Fig. 6).

In LUT methods, cubic, tetrahedral or non-linear interpolation is performed to calculate the corresponding XYZ values for the RGB grid. This requires that the ground truth/measured values be placed at a regular interval in the grid [9]. Since, in our application, it is not possible to have the ground truth data points (928 data points extracted from the calibration target) to be positioned equidistant across the LUT, interpolation techniques can not be used. The XYZ values of the corresponding grid points in Fig. 6 were calculated by linear LSA method using n nearest calibration data points. n value of 30 is found to be suitable for our application.

For *color correction*, images pass through the same image processing pipeline (Fig. 4). Afterwards, the chosen color transformation among the aforementioned methods is applied. When using LUTs, the out-of-gamut colors are mapped via absolute gamut mapping i.e., the closest point on the LUT gamut is selected.

D. Evaluation

Three different targets (shown in Fig. 7) were selected to evaluate the accuracy of color correction:

- 1) IT8/7.4 CMYK chart (1,620 colors)
- 2) IT8/7.3 CMYK chart page 1 (468 colors)
- 3) IT8/7.3 CMYK chart page 2 (460 colors)

A total of 2,548 colors were used for evaluation. It should be noted here that although the test charts 2 and 3 are the two-paged version of the calibration target described in Section III-A, they are printed and measured independently of the calibration/training chart. The results were evaluated in terms of color difference (CIE ΔE_{00}) values between the calibrated camera colorimetric values and the ground truth values measured spectrally. We used the mean color difference to evaluate the accuracy of the calibration algorithm. However, in some cases the mean could be skewed due to few

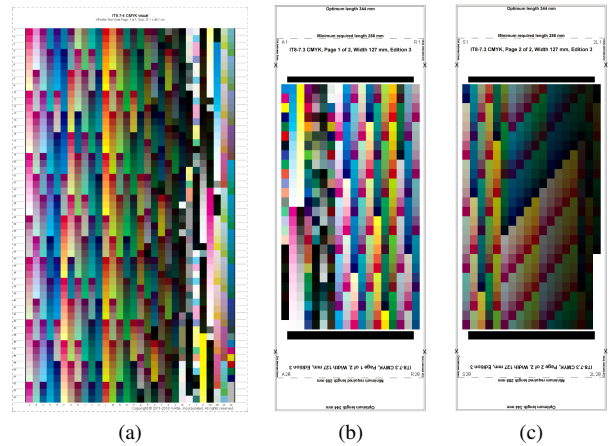


Fig. 7: (a) Test chart 1 (b) Test chart 2 (c) Test chart 3

abnormally high color difference values. To get a better idea of the results, we also looked at the median values of the color differences to see the general trend of the color difference while testing. The maximum color difference is also a good measure to check if some of the colors have an abnormally large color difference value.

In addition to mean, median, and max, it is useful to look at the percentage of samples that fall below certain color difference thresholds. In printing industry, a color difference of less than 2 is considered ideal and is almost imperceptible by the human eye. However, it is quite difficult to achieve, and a color difference of less than 5 is considered acceptable [13]. We have calculated the percentile rank of samples that fall below these threshold values to evaluate the quality of the tested calibration algorithms.

IV. RESULTS

For each of the mapping algorithms described in Section III-C, the predicted values were compared with the measured ground truth data. In TABLE I, the results for all 2,548 color patches are summarized quantitatively through mean, median, maximum of the color differences, and the percentile rank of data points with color differences less than 2 and 5 respectively.

Based on the results in TABLE I, we can infer that the 3D LUT method provides the best result based on the lowest mean color difference and highest percentage of color patches with both color difference less than 2 and 5. The results from HPCC and LUT 2.5D methods are quite similar which is logical since both of them are based on establishing relationship between rg chromaticity values and the corresponding XYZ values. Test charts 2 and 3 before and after color correction by LUT 3D method are shown in Fig. 8.

An interesting observation was made when the color charts used for evaluation (mentioned in Section III-D) were analyzed separately. The results of individual test chart evaluation for the two methods which performed the best (LUT 3D, LUT 2.5D) are presented in TABLE II. Test chart 1 performed better with LUT 3D as compared to LUT 2.5D with lower mean, median, and maximum color difference values and higher percentile rank for both the color difference thresholds. Test chart 2 performed well for both LUT 3D and 2.5D but its performance with LUT 3D is exceptionally good with 96% of the total colors with color difference less than 5 units. Test

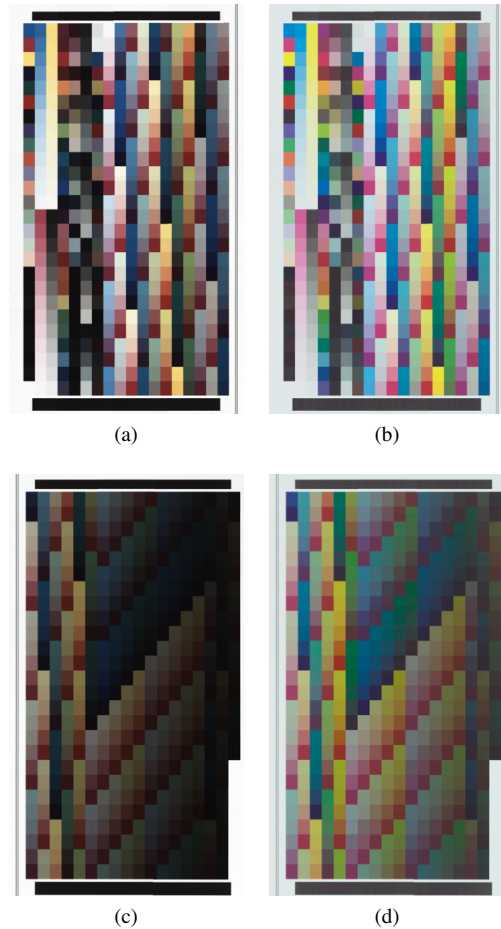


Fig. 8: (a) Test chart 2 before color correction (b) Test chart 2 after color correction (c) Test chart 3 before color correction (d) Test chart 3 after color correction

chart 3 performed better with LUT 2.5D as compared to LUT 3D.

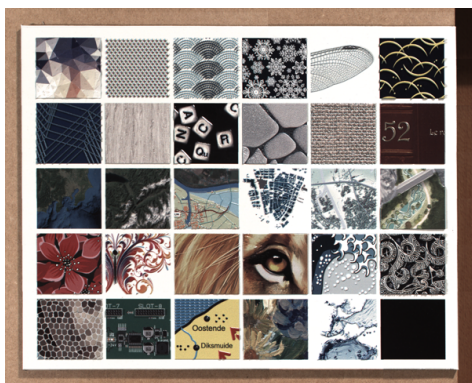
It should be noted that test chart 2 has predominantly lighter and high chromaticity colors while test chart 3 has predominantly darker low chroma colors. The average L^* value for test charts 1, 2, and 3 was calculated to be 48, 60, and 40 respectively. Considering results from test charts 2 and 3, we can say that the performance of LUT 3D is better suited for colors with higher lightness values and LUT 2.5D performs well for darker low chromatic colors. The chromaticities of

TABLE I: Overall results

Method	Mean (ΔE_{00})	Median (ΔE_{00})	Percentile Rank ($\Delta E_{00} \leq 2$)	Percentile Rank ($\Delta E_{00} \leq 5$)	Max (ΔE_{00})
Linear LSA	4.3	3.46	20.92	68.01	16.76
Polynomial 2nd degree LSA	3.23	2.83	29.71	84.03	17.76
Root polynomial 2nd degree LSA	3.67	2.9	29.43	77.59	18.66
Cube root polynomial LSA	3.7	2.86	30.71	82.86	17.53
HPCC (9 hue segments)	3.38	2.63	31.95	81.67	18.08
LUT 3D	3.07	2.72	32.06	86.62	17.77
LUT 2.5D	3.43	2.76	30.14	81.5	18.05

TABLE II: Individual test chart results

Test chart	Method	Mean (ΔE_{00})	Median (ΔE_{00})	Percentile Rank ($\Delta E_{00} \leq 2$)	Percentile Rank ($\Delta E_{00} \leq 5$)	Max (ΔE_{00})
Test chart 1	LUT 3D	3.24	2.86	24.32	86.73	16.06
	LUT 2.5D	3.87	3.19	21.79	76.91	18.05
Test chart 2	LUT 3D	1.78	1.37	70.3	96.58	15.02
	LUT 2.5D	2.54	1.8	53.63	89.32	13.88
Test chart 3	LUT 3D	3.83	3.6	20.43	76.09	17.78
	LUT 2.5D	2.85	2.35	35.65	90	12.92



(a)



(b)

Fig. 9: An example image (a) before and (b) after color correction

darker colors are more ambiguous and thus a method more sensitive to slight changes in chromaticity rather than the absolute values is better suited for correct color prediction in those regions.

V. CONCLUSION

In this paper, we have presented a complete framework for acquiring, processing and color correcting images in a non-laboratory setting. Methods for compensating non-uniformities and distortions introduced by the environment and equipment were recommended. Several color calibration and correction methods were tested and the results were analyzed to identify

the most suitable method for the application. The lookup table based methods were found to be the most accurate. Lookup based methods work best with large amount of ground truth data. Conventional color charts such as Macbeth ColorChecker can not provide enough training data to form a well-performing LUT.

In the light of the results presented in Section IV, more investigation needs to be done regarding the different behavior of mapping function in lighter and darker regions of the color gamut. A hybrid lookup table combining the better performing aspects of 3D and 2.5D LUTs could be a good solution to perform more robust and accurate color correction.

REFERENCES

- [1] N. Ohta and A. R. Robertson, *Colorimetry: Fundamentals and Applications*, 2006.
- [2] J. Jiang, D. Liu, J. Gu, and S. Susstrunk, "What is the space of spectral sensitivity functions for digital color cameras?" in *Proceedings of IEEE Workshop on Applications of Computer Vision*, 2013.
- [3] B. K. Horn, "Exact reproduction of colored images," *Computer Vision, Graphics and Image Processing*, 1984.
- [4] G. Hong, M. R. Luo, and P. A. Rhodes, "A Study of Digital Camera Colorimetric Characterisation Based on Polynomial Modelling," *Color Research & Application*, 2001.
- [5] F. Fang, H. Gong, M. Mackiewicz, and G. Finlayson, "Colour Correction Toolbox," in *13th AIC Congress*, 2017, pp. 13–18. [Online]. Available: https://ueaeprints.uea.ac.uk/65098/4/Colour_Correction_Toolbox.pdf
- [6] G. D. Finlayson and G. M. Johnson, "Extended Linear Color Correction," in *Color Imaging Conference*. Society for Imaging Science and Technology, 2016, pp. 168–173.
- [7] E. A. Fedorovskaya, H. De Ridder, and F. J. Blommaert, "Chroma variations and perceived quality of color images of natural scenes," *Color Research and Application*, 1997.
- [8] C. Find-Andersen and J. Y. Hardeberg, "Colorimetric Characterization of Digital Cameras Preserving Hue Planes," in *IS&T/SID 13th Color Imaging Conference: Color, Science, Systems and Applications*, 2005, pp. 141–146.
- [9] P. Hung, "Colorimetric calibration in electronic imaging devices using a look-up tables model and interpolations," *Journal of Electronic Imaging*, vol. 2, no. January, pp. 53–61, 1993.
- [10] M. R. Pointer, "The Gamut of Real Surface Colours," *Color Research & Application*, 1980.
- [11] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, 2002.
- [12] J. Y. Bouquet, "Camera calibration toolbox for MATLAB," *High Speed Vision System & Object Tracking*, 2010.
- [13] "The Print Guide: Tolerancing color in presswork - CIE L*a*b* and DeltaE." [Online]. Available: <http://the-print-guide.blogspot.com/2010/04/tolerancing-color-inpresswork-cie-lab.html>