A COLOR BLENDING MODEL AND A COLOR CORRECTION ALGORITHM FOR ADDITIVE OPTICAL SEE-THROUGH DISPLAYS

SRIKANTH KIRSHNAMACHARI SRIDHARAN

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada

ABSTRACT

Optical see-through display (OSTD) is a transparent digital display which simultaneously gives access to the digital contents and the real world objects behind it. Additive optical see-though display is a hardware subtype of OSTD which has its own light source to create the digital contents. In Additive OSTD, light coming from background objects mixes with the light originating from the display causing what is known as the color blending problem.

The work in this thesis provides a solution to the color blending problem. In order to understand the problem, this thesis first presents a new color blending model for additive OSTD based on two display induced distortions: the *Render* distortion and the *Material* distortion. A new method called *Binned Profile* (BP) method which accounts for the *render* distortion is developed to predict the blended color, when applied on the color blending model. BP method is validated with other known methods and is shown to be the most accurate in predicting the color blends with 9 just noticeable differences (JND) in worst case. Based on the BP method, a new color correction algorithm called BP color correction is created to solve the color blending problem. BP-color correction finds the alternative digital color to counter balance the blending. The correction capacity of various digital colors were analysed using the BP color correction approach. BP color correction is also compared and proven to be better than the existing solution. A quicker version of the correction called quick correction is also explored. The thesis concludes with an exploration of the *material* distortion, explains the limitations of BP-correction, provides design recommendations .

PUBLICATIONS

Some ideas and figures in this thesis have appeared previously in the following publications by the author:

Srikanth K Sridharan, Juan David Hincapié-Ramos, David Flatla,Pourang Irani Color Correction for Optical See-Through Displays Using Display Color Profiles. In *19th ACM Symposium on Virtual Reality Software and Technology (VRST2013)* , 2013.

IMAGES USED IN THIS THESIS

Images used in this thesis as background samples are photos captured by the author.

The foreground images were from images shared in Google images by Google glass, Google maps, Google Play store - Pocket Casts and Epson Moverio as example of their user interface.

ACKNOWLEDGMENTS

This thesis and research has been part of my life for more than two years now. I have to thank many people who have helped me along the way and I am grateful to have made their acquaintance.

The first person I would like to thank is Dr. Juan David Hincapié-Ramos. Without whom I would not have been able to finish the thesis in the time frame I did. I learned a lot from Dr. Juan David in the past two years. I thank him for his patience, guidance, support and motivation. Under him I have learned basics of research from writing a good paper to collect and analyse data. Over the years I have come to appreciate the meaning of his motto "convince me!!" and "Build it first". I can for sure say things he has taught, helped me to be a better person both in my personal and professional life. More than anything I thank him for being a honest friend.

I thank my supervisor Dr. Pourang Irani for his confidence in my ability and unending support he has given towards improving my research skills. Under his guidance I have learned a lot about what the HCI research community is about and how to value good research. Without Dr.Irani's constant support and his ability to get things need for this research, this thesis would not have been possible.

I would like to thank my dad K.Sridharan, whose only dream is to see his children prosper. Without his efforts, I would not be the person I am today neither this research would have been possible. I would also like to thank my sister and grandparents. I strive every day to be a person worthy of their unconditional love.

I thank all my colleagues in U of M's HCI lab for their enduring support. I would specially like to thank Jango Guo and Levko Ivanchuk for their help with my research. I also thank my thesis committee members Dr.Arkady Major and Dr. Neil Bruce for their suggestions in making my work into a quality one.

I would also like to thank Brahm Tewatia of Folio Instruments Inc. and Jay M. Catral of Konica Minolta for their timely help in making CS-200 colorimeter available for me to conduct my research.

At-last I would like to thank my special someone. You have equal share in this success.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND TERMS

CIE International Commission on Illumination

CIE-XYZ CIE 1931 XYZ color spaces

CIE-LAB CIE 1976 (L*, a*, b*) perceptually uniform color space

Color distance Euclidean distance between two colors in CIE-LAB space

OSTD Optical see-through display

BP Binned profile

QC Quick correction

DM Direct method [27]

CAT Chromatic adaptive transportation [25, 16]

JND Just noticeable difference between two color values in CIE-LAB color space [18] 2.3

HMD Head mounted Displays [23]

SLM Spatial light modulation [3]

UI User Interface

RGB Red Green Blue - color space

sRGB	Standard RGB Color space
AR	Augmented reality
VR	Virtual reality
LUT	Look UP Table
LCD	Liquid crystal display

1

INTRODUCTION

Optical See-through Displays (OSTD) are transparent digital displays which allow simultaneous view of both digital content and the physical background . OSTD come in two major hardware subgroups, namely additive and subtractive OSTDs. "Additive" OSTDs have their own light source to create a pixel. Display technologies such as transparent organic light emitting diode (T-OLED) and projection based OSTD belong to this hardware subtype. Subtractive OSTD do not have their own light source. They filter white light from an external source to create a pixel. They need a powerful and bright white light source active all time to work. Display technology such as Transparent LCD is a subtractive OSTD. Wider adaptation of OSTD in retail consumer electronics such as Google Glass, Epson Moverio and Lenovo S800 phones has pushed additive OSTD to mainstream use.

One major issue with "Additive" OSTD is that light coming from real-world objects mixes with the light emitted by the display, which changes the digital content's color for a human observer. This problem in is known as color blending [4]. Color blending can be seen in Figure 1 where yellow digital color changes towards red and blue on blending with the red and blue background, respectively. Color blending negatively affects the legibility and color coding of digital content, compromising the general usability of OSTD devices. There are several approaches to solve the color blending problem, such as use of SLM Spatial light modulation (SLM) devices [3, 12, 10, 28] which completely blocks out the background or Digital interface relocation based on the background [26]. The major disadvantage in SLM is the need of extra hardware enhancement and in digital interface relocation there might be constant shift of digital contents on the display. The work in this thesis provides a different solution as it aims to avoid both these disadvantages.

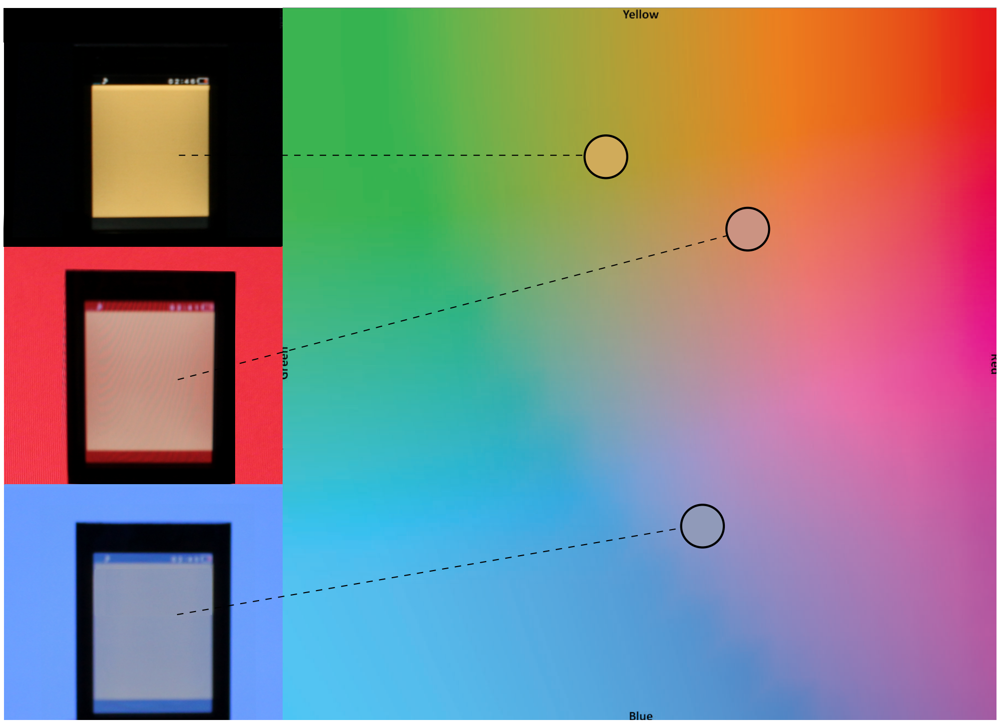


Figure 1: (Image best seen in color) Left Image: OSTD display shows a yellow rectangle on (1) black, (2) red and (3) blue backgrounds. Right Image: blend colors in LAB color space.

Another solution to color blending is called "Color Correction". Color correction involves in picking an alternate digital color which, upon blending with the current background comes, closer to the digital color intended by the UI designer. Weiland et al. developed a color correction system for OSTD which is based on subtraction digital color's RGB value with the background color's RGB value. The major drawbacks of Weiland et al.'s work is correction achieved via RGB subtraction is highly inaccurate.

The work in this thesis also focuses on developing a color correction system but moves away from simple RGB subtraction. Work in this thesis shows that the color correction depends on color blending and a accurate color correction system needs to know how colors blend in OSTDs. This thesis introduces a new color prediction model, to predict how colors blend in an "Additive" OSTD. A novel method called *BP method* is created to implement the color prediction model. Once BP method was shown to be accurate in predicting the color blends, color correction algorithm called *BP color correction* is created to counter balance color blending. Standard set of colors were used and their correction capacity were tested on three different "Additive" OSTDs. A quicker version of color correction algorithm is also presented. Further research implications of this work and possible future research are discussed.

1.1 MOTIVATION

Gabbard et al. [4] demonstrated the significant change in a digital color under the influence of a physical background. This change in color causes perceptual problems in field of augmented reality (AR) as explained by Kruijff et al. [13] and it is more pronounced in outdoor use of OSTD as shown by Keer et al. [9]. The users in such case had to use their hand to block off the background to have legible digital content in their HMD (Head mounted Displays) [23]. Work by Kiyokawa et al. [11] tries to counter this by dynamically increasing digital contents brightness, however as shown by Keer et al. [9] this can not be achieved for outdoor background colors.

As a solution to color blending Tanaka et al. [26] proposed a user interface layout system that relocates digital content to darker background regions in the display. However this method may lead to constant relocation of the interface for a dynamic background and will not work in conditions where every background color is bright.

Color blending also affects the general usability of OSTD in the domain of spatial augmented reality. Spatial augmented reality deals in projecting digital contents on top of a physical object , such as a textured screen [22], textured wall [2] or a three dimensional physical object [2]. Color blending in spatial augmented reality affects the effective occlusion of physical objects by the digital content. Without effective occlusion, the virtual object stands out as unreal. The negative effects of this is explained in the works by Cakmakci et al. [3]. Sekuler and Palmer [24] showed that the virtual context and usefulness of a VR system is affected by background's physical color, as it creates translucent digital content. Spatial light modulation (SLM) device [3, 12, 10, 28] attached to head-mounted displays to prevent color blending and achieve effective occlusion. SLM prevents color bending by completely blocking out the background in the needed regions.

This thesis differs from previous solutions as it does not aim to change the location of user interface elements or to add new hardware components to the OSTD. This work's prime focus is on preserving the displayed color, by counter balancing the effect of color blending.

1.2 BACKGROUND

Color correction consists of finding an alternate digital color which, on blending with the background's color, comes closer to the desired digital color. To find an alternate digital color to counter balance color blending, it becomes imperative to understand how colors blend in OSTD.

Gabbard et al. [4] quantitatively documented color blending in a OSTD by building an experimental test-bed. They measured six common outdoor background colors on 27 digital colors on the edge of the sRBG gamut. The six common outdoor colors chosen were foliage, brick, sidewalk,pavement, white and no background. Their results showed that high intensity backgrounds changed all display colors by pulling them towards white and backgrounds of different hues pull all colors toward them. Gabbard et al. [4] also formulated a color blending equation (See Equation 1.1) which lists out the factors which contribute to change in digital color's perception. This equation represents color blended and explains the color perceived by a user (CP) as a function of the light source (L1), background (B) object's reflectance (RF), the light emitted by the display (L3), the interaction of both L1 and L3 in the display (AR_D), and the human perception (HP). However Gabbard's equation does not explain how these factors interact to create the blended color, rather it simply states the factors involved.

$$CP = HP(AR_D(L_3, RF(L_1, B))) \tag{1.1}$$

Taking equation 1.1 as the starting point, this thesis proposes a color blending model for additive OSTD. In the model formulation, the light and reflectance of the background (RF(L1,B)) are taken as the one unified factor called "background color" (BC). Color addition accounts for the interaction of light in the displays (AR_D). The model accounts for human perception (HP) by using the CIE XYZ and LAB color spaces (later explained in the this chapter) which are based on human color perception. The color blending model for "Additive" OSTD is formulated as :

$$BlendedColor = f_{render}(DC) + f_{material}(BC) \tag{1.2}$$

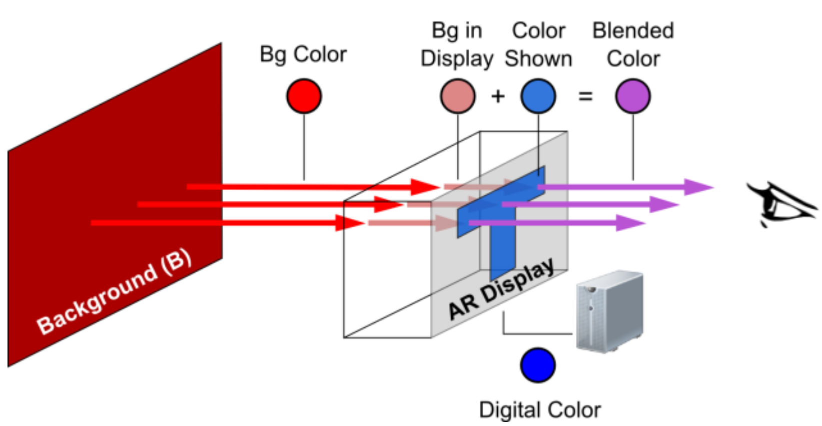


Figure 2: (Image best seen in color)Color blending concept with the *render* (Color shown) and *material* distortions (BG in Display) for digital and background colors.

This model predicts the resulting color blend by taking into account two color distortion functions: the *render* distortion function (f_{render}) and the *material* distortion function ($f_{material}$). f_{render} explains how a particular display renders colors and $f_{material}$ explains how the display material (acrylic or glass) changes background colors. Interaction between these functions are shown in the Equation 1.2. Figure 2 shows how the *render* distortion changes the digital color blue (digital color to color shown) and the *material* distortion changes the background color red (Bg color to Bg in display).

Figure 3 left shows how the *Render* distortion results in the change in displayed digital color red (#FF0000) for the three different OSTD. Figure 3 right shows how *material* distortion results in the change of the background color foliage in the various OSTD material. It can be seen that there is a significant change in the displayed color due to *render* distortion when compared with change in the background color due to *material* distortion.

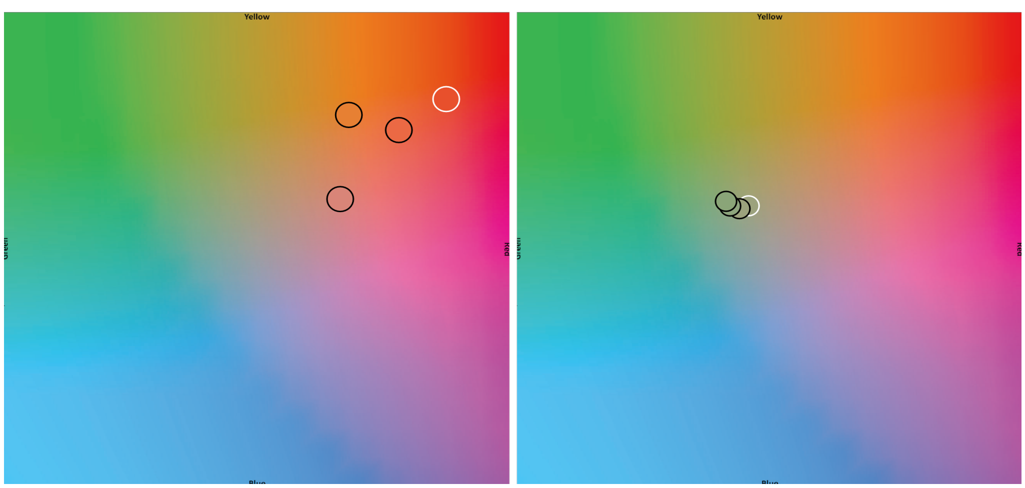


Figure 3: (Image best seen in color) Left: Render distortion - the color red (#FF0000) (white border) and as displayed by three OSTD. Right: material distortion - foliage color (white border) and as it is seen through two OSTD material.

1.3 COLOR SPACES

Color space is a mathematical model which maps colors visible to humans into a three dimensional space. Color space are usually used to represent and reproduce colors. There are two standard color used in this thesis apart from RGB color space.

1.3.1 *CIE XYZ*

CIE 1931 XYZ color space is the first mathematically defined color space. It was created by the International Commission on Illumination (CIE) in 1931. CIE XYZ is based on human vision and how a human observer sees color. The color addition used in this thesis was based on CIE XYZ color addition.

Let color 1 and color 2 be two colors, let color 3 be the blend of color 1 and 2 , then formula 1.3 gives the color value for color 3.

$$
\begin{aligned}
Color3X &= Color1X + color2X \\
Color3Y &= Color1Y + color2Y \\
Color3Z &= Color1Z + color2Z
\end{aligned} \tag{1.3}
$$

1.3.2 *CIE LAB*

CIE LAB color space is a uniform color space with almost equal spread to all color values , based on non-linearly compressed CIE

XYZ color space coordinates. The color difference and hue difference calculation done in this work are based on CIE LAB.

Euclidean Distance and Hue Difference Calculation

Euclidean Distance Calculation is the difference between two color values in CIE -LAB space, if the value is found to be less than 2.3 then two colors are found to appear same for human observer. If color 1 is represented by L1,A1,B1 and color 2 represented by L2,A2,B2 then the formula for this is given ny the equation 1.4 and formula for basic hue difference is given by the equation 1.5.

$$ColorDistance(\Delta E) = \sqrt{(L1 - L2)^2 + (A1 - A2)^2 + (B1 - B2)^2} \tag{1.4}$$

$$\begin{aligned} XD &= \sqrt{A2^2 + B2^2} - \sqrt{A1^2 + B1^2} \\ ADifference &= A2 - A1 \\ BDifference &= B2 - B1 \\ HueDifference(\Delta h) &= \sqrt{ADifference^2 + BDifference^2 - XD^2} \end{aligned} \tag{1.5}$$

1.4 SCOPE

Two projection based OSTD and one T-OLED are used in this research to cover different types of additive OSTD. The *render* distortion function is key to have accurate color blending model. To characterize the *render* distortion, a display profiling technique called Binned Profile(BP) is created. This BP was used in my model to predict how colors blend. The color blend prediction accuracy of BP method was tested against other known methods for estimating how a display renders color; the direct method (DM) and three chromatic adaptation transformation (CAT) methods [25]. The direct method ignores the *render* distortion (called "trivial correction" by Weiland et al. [27]). The CAT methods use known color transformation matrices based on the brightest white value on OSTD. *Material* distortion was accounted for by objective measures of background colors as seen through the display material. On experimental validation, the BP method when compared to other methods was found to have a low error rate - with zero just noticeable differences (JND) in best case and within nine-JND in the worst case. Result also shows that accounting for *render* distortion is essential for color prediction model.

A color correction algorithm was developed based on BP method. Using this color correction algorithm, correction capacity of large set of digital color were studied. A color correction algorithm with faster execution time was also developed. A possible way to handle *material* distortion is also explained. Limitations of BP color correction and its design implications are discussed in detail.

1.5 CONTRIBUTION

The contributions of this thesis are:

- A color bending model for "additive" OSTD.
- Binned profiles to address the *render* distortion.
- BP based color correction for "additive" OSTD.
- Exploration of a color correction algorithm with faster execution time.
- Explanation of a possible solution to *material* distortion

2

RELATED WORK

Color correction is the core area of this thesis, this chapter will concentrate on existing works on color correction in AR and VR. Other ways to solve color bending such as SLM and interface layout switching were already discussed in section 1.1.

2.1 SPATIAL AUGMENTED REALITY

Researchers in the field of spatial AR, in particular projector-based AR have studied color correction as a way to enable projections on textured or colored surfaces. Nayar et al. [22] use a camera-based radiometric calibration model. This model computes the relation between the digital image and the projected blend on a textured surface. Their approach requires a calibration phase where known digital patterns are projected on the projection surface and the resulting blend is processed to obtain compensation matrices. This method works only on a static known background, if the background texture or color is changed the whole process of calibration needs to be re-done. This method only allows a limited number of colors to be corrected, as various factors such as the power of the projector and the projector's angle alter the display's color gamut. Bimber et al. [2] extended the range of projectable color by using a transparent film on top of the textured surface and multiple projectors. Their model takes into account the reflectance and absorption of the digital color by the projection surface. Absorption by the projection surface makes black background as their worst case scenario as it absorbs most of the projected light making it harder to distinguish the digital contents. All these works assume that the textured surface is in a dark room. Grossberg et al. [5] extended the radiometric model of correction to include ambient light. They did this by measuring the reflectance of ambient light from the projection surface and counter balancing that value in their model. The primary objective of these works was to achieve content legibility on a textured surface. While these works are in the device dependent sRGB color space, others achieved higher correction accuracy by working on the device independent CIE XYZ color space [1, 20]. Work by Ashdown et al. [1] deals primary in CIE XYZ space, the purpose of the work was to achieve accurate color correction on textured surface. Work by Menk et al.[20] extended this to projection on physical objects. Menk et al. also concentrated on achieving accurate color correction on a single known background color. They created an sRGB look up table (LUT) to define how each projected color pixel appears on the physical object. They then used this LUT to look up the color they wanted. Their solution works only for a single static background color, each time the background changes the LUT needs to be recreated.

2.2 Optical see-through display

The only existing work on color correction of OSTD is by Weiland et al. [27]. Weiland et al used color correction to improve content legibility and contrast improvement. Their system is based on Bimber et al. [2] with a high dynamic range (HDR) camera on top of the display to capture the background. Correction is achieved by subtracting the camera-captured background color from the display color using pixel shader. They used simple RGB subtraction where the foreground color's RGB value was subtracted from background color RGB value as shown in equation 2.1. Weiland et al's color subtraction ignores both the render and material distortions and yields colors outside the sRGB space for most backgrounds.

$$CorrectedColor = ForegroundRGB - BackgroundRGB \tag{2.1}$$

This research is inspired by the same line of work, but moves away from color subtraction. This thesis focuses on the actual colors an OSTD can show and treats background colors as seen through the display material. BP-based color correction uses a best-fit approach to find the display color which, upon blending with the background, comes closest to the desired display color. Unlike Weiland et al to achieve high correction accuracy, this work takes inspiration from works in spatial augmented reality [1, 20] and uses the device independent CIE XYZ and CIE LAB color spaces. This study was also extended to both projector-based and T-OLED (Transparent Organic

Light Emitting Diode) displays, and results are presented quantitatively. None of the above mentioned work analysed the colors beyond what their system can correct, however this work also explores colors that can and cannot be corrected. This is important in order to understand how color work in OSTD.

3

HADWARE AND SETUP

3.1 OSTD HARDWARE

"Additive" OSTDs do not depend on any external light source to create digital content. This gives additive OSTD the flexibility to be used in various form factors such as head-mounted display (HMDs), window-size, desktop-size and mobile-size displays; as they do not need a large back-light support. In this thesis the experimental test-bed consists of three such "Additive" OSTDs: two projector-based window displays and one transparent OLED mobile display.

3.1.1 *Projection Based*

Projection based OSTD has a transparent film on which digital contents can be created using a digital projector. The transparent film used in this study is called Lumisty [17]. Two different projectors of varying lumens were used to project digital contents onto the Lumisty. The first projector is an Epson 1705 with 2200 lumens, henceforth will be referred as "p2200". The second projector is an Epson $VS35OW$ with 3700 lumens, henceforth will be referred as "p3700". The set-up of both the Projection based OSTD is shown in figure 4. These projector-based displays use a 3 mm transparent acrylic surface covered with a Lumisty MFZ 2555 film and one of the two projectors at 40°. Lumisty MFZ 2555 [17], is a view-control film that diffuses light incident on it from blocking angles(+55 to +25 and -25 to -55) to the normal as shown in the Figure 5. While light can pass through it in all other angles. OSTD set-up was created by placing the projector in the blocking angle, the digital contents were projected on to the screen. While the background can be seen with the digital contents from any other angle.

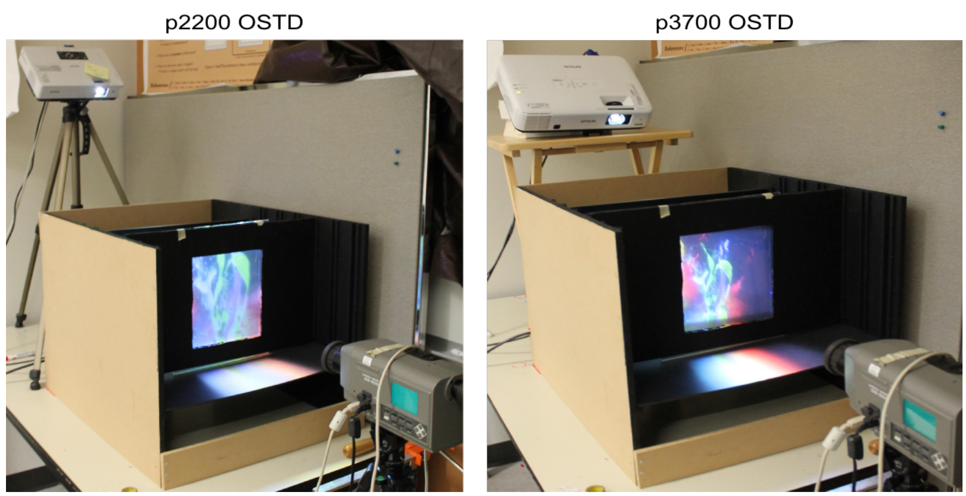


Figure 4: Projection based OSTD setup which shows prjector at 40° to the Lumisty, creating visible digital content. Left Image: p2200, Right Image: p3700

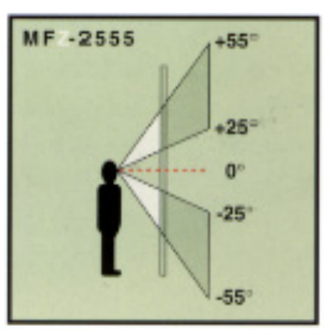


Figure 5: Lymisty MFZ 2555 view angles

3.1.2 *T-OLED*

T-OLED OSTD is a digital display where all pixels were made up of transparent organic light emitting diodes. For this thesis the transparent OLED display used was a Lenovo S800 phone [14] with resolution of 240x320 at 167 pixels per inch. This display will henceforth be referred as T-OLED. The T-OLED display is covered in 9 mm thick acrylic. Figure 6 shows the transparent nature of the T-OLED display.

Figure 6: T-OLED used in this thesis is the screen of Lenovo S800 mobile phone [14]

3.2 EXPERIMENTAL SET-UP

Figure 7C shows our experimental test-bed built (i) to generate various background colors, (ii) to show display colors on multiple OSTDs, and (iii) to measure color blending. Various background colors were generated by using a Dell U2312HM LCD display calibrated at the standard D65 white point, the standard outdoors lighting condition. Both the projection based and T-OLED OSTDs were kept at distance of 20 cm from a LCD monitor, as shown in Figure 7A and B. A colorimeter (Konica Minolta CS-200) was placed 20 cm in front of the OSTD pointing to the display's center. This colorimeter was used to measure the color blend between the color displayed on the OSTD and color from the background LCD. A tarpaulin (opaque and heavy-duty cloth) was used to block influence of external lighting as shown in Figure 7D (Dark Cave).

The experimental setup used the ColorChecker Color Rendition Chart [19] colors as backgrounds. ColorChecker colors are well spread out inside the color space and represent colors of everyday natural objects like sky, skin and foliage. Figure 8A shows the difference between theoretical background colors and how the test-bed produced them. This test-bed differs from previous systems [4] which prioritize the capacity to obtain background colors close to as seen in nature. Even though usage of LCD limited the background colors to fall within the sRGB gamut, the test bed was designed to produce a wide variety of background colors. The LCD monitor reproduced 23 ColorChecker colors which were inside sRGB gamut.

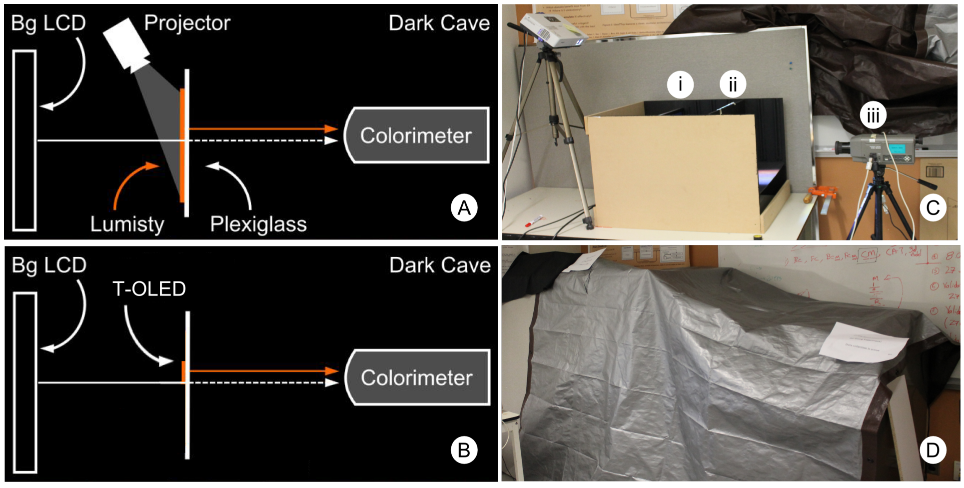


Figure 7: Figure A: Set-up for both p2200 and p3700
Figure B: Set-up for T-OLED,
Figure C: Shows the actual set-up with (i) LCD monitor used to simulate the background,(ii) Holder which holds the OSTDs in place and (iii) Colorimeter,
Figure D: Shows the Dark Cave created with a tarpaulin

The Same controlling computer connected the various displays and the colorimeter.

The notations of the Commision Internationale de l'Éclairage (CIE) color model were used to examine the resulting color blends. The set-up measured the colors in CIE 1931 XYZ color space for addition required by the color blending model. The XYZ color space is not a perceptually uniform color space as it resembles the working of the human visual system, which is more sensitive to greens. This causes color green to have more spread than blue and red hues in this space. The set-up calculated the perceptual difference between colors using CIE 1976 LAB color space, a perceptually uniform color space. The distance between a color and its shift when it blended or the distance between a prediction and the measured blend, were measured in LAB.

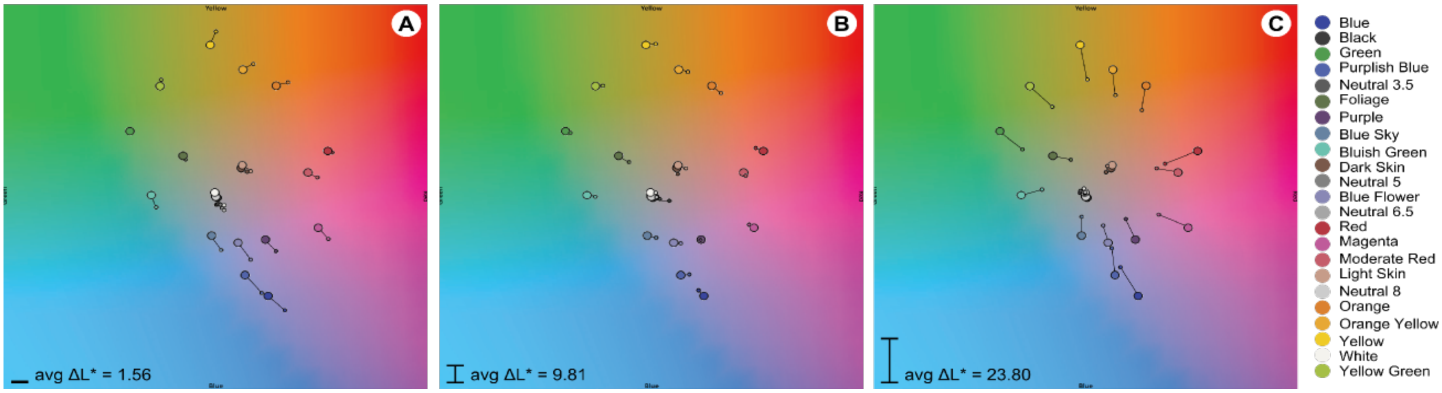


Figure 8: (Image best seen in color) ColorChecker bg colors as (A) shown by the background LCD, (B) as seen through the p2200 and p3700 displays, and (C) as seen through the T-OLED display. Bigger circles = original color. Small circle = measured color.

The set-up used Konica Minolta's CS-200 luminance and color meter at 0.2° (standard observer angle) for color measurement. The CS-200 measures colors in the XYZ color space. In order to convert these values into normalized LAB as explained in Gabbard et al. [4], the colorimeter measured the brightest white point of the all three displays involved. For both p2200 and p3700 OSTD XYZ values of the brightest white points of the Lumisty surface were measured at 5 different points: one near the each of the display's four corners and one in the center. For both OSTDs all measurements of the white point remained constant. After calibrating the background LCD to D65 (measured at 0.9504, 1, 1.0888) we measured the following two combinations of the white point per display and recorded the average of 100 measures per combination(see Table 1):

1. See-through showing white and bg LCD turned off.
2. Both see-through and bg LCD showing white.

Table 1: White points for all three displays.

		p2200	p3700	T-OLED
No BG	X	0.2656	0.9504	0.3833
	Y	0.2822	1	0.3950
	Z	0.4810	1.0888	0.3699
White BG	X	0.9504	0.9504	0.7248
	Y	0.9900	1	0.7599
	Z	1.0888	1.0888	0.7273

4

COLOR BLENDING MODEL

In order to build a reliable color correction system, it is necessary to have an accurate model of color blending. Equation 1.2 explains the color blending model. Section 1.2 explains two distortions given in this equation namely, the *render* and *material* distortion. This thesis argues: "*Color blending in additive OSTD is based on two factors : the render and the material distortion*". This section explains how to handle these two distortions. This section also shows how this model holds up against a color prediction which, ignores both these distortions. The $f_{material}$ function is handled by the use of colorimeter to capture the background through the display as shown in figure 8.

This chapter presents a methods called Binned Profiles to account for the f_{render} function. Binned Profiles method receives a color the display wants to show as the parameter and returns a approximate color the display actually shows. This chapter will also shows that accounting for f_{render} holds the key to the accuracy of the color blending prediction in OSTD.

4.1 BINNED PROFILE METHOD

This thesis proposes binned-profiles (BP), a display profiling technique which divides the continuous sRGB color space (over 16 million colors) into a smaller set of 8390 perceptually different color bins. To create the color bins, sRGB space was converted into the CIE LAB by dividing it into bins of size $5 \times 5 \times 5$. This was based on the approach proposed by Heer and Stone [7], of the bins to limit the color space into a finite number of colors. They used bins to create a color naming system . The $5 \times 5 \times 5$ bin in CIE LAB space ensures all colors inside a bin are within one Just Noticeable Difference. 1 JND is approximately equal to 2.3 Euclidean distance in CIE LAB space and all colors within 1 JND are perceived as the same color by a human observer [18]. Figure 9 A-B shows all the sRGB gamut in the CIE LAB color space and $5 \times 5 \times 5$ bin values (with one representative color for all the other colors within the same bin).All three OSTD displayed each of representative colors one after the other while the colorimeter measured them. The system converted the measured XYZ values into CIE LAB color space, using the reference white points given in Table 1. The background LCD was turned off while these measurements were made to produce a no background condition. The system created a color profile for each display based on these measurements. Figure 9 C-E presents the display profiles, p3700 was found to almost be matching color capacity of sRGB (C), while p2200 (D) and T-OLED displays (E) exhibited significant reductions of color capacity in comparison to sRGB.

BP method uses BP of a particular display to account for f_{render} function (see Algorithm 1) in the equation 1.2 to predict the color blend. To account for how a display color changes, BP method first

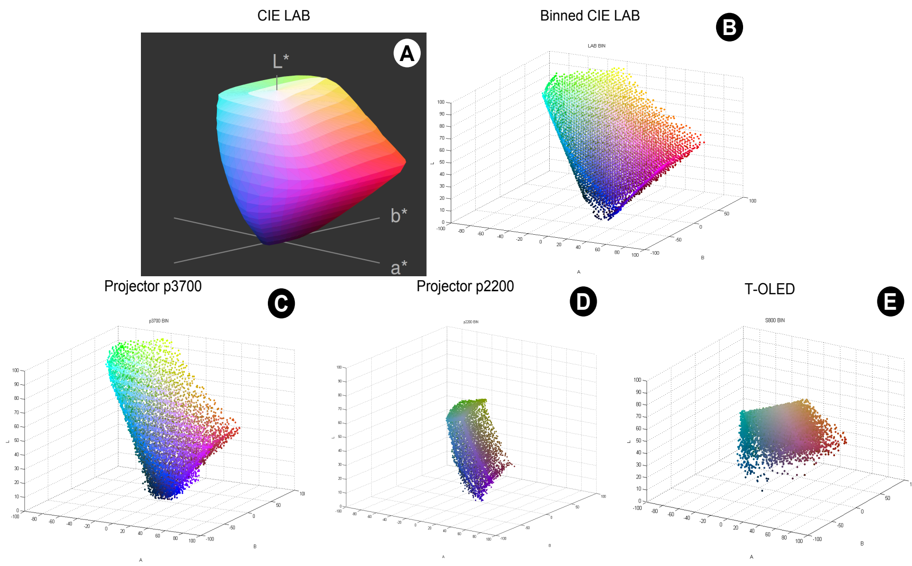


Figure 9: (Image best seen in color)(A) sRGB gamut on the LAB color space, (B) the binned gamut, and the binned profile for the (C) p3700 and (D) p2200 projector-based displays, and for (E) for the T-OLED display.

Algorithm 1 Binned-Profile based prediction algorithm

procedure BP-PREDICTION(*Display*,*Foreground*,*Background*)
  $BinForeground = findBin(Foreground)$
  DisplayForeground=*lookup*(*Display*,*BinForeground*)
  Prediction=*addXYZ*(*DisplayForeground*,*Background*)
return *Prediction*
end procedure

converts a sRGB color to its closest bin in LAB space (findBin). This is achieved by converting the sRGB value into its corresponding LAB value, then rounding of L, A and B values to their nearest multiple of 5. BP method then uses this bin value as a key to look up in the profile lookup table(lookup function). The measured color linked with the key is the one the display actually shows (Color Shown in Figure 2). BP method then uses this color in the color blending model (equation 1.2) and adds it to the background to obtain the color blend. The addition is done in CIE XYZ color space (addXYZ function).

4.2 DIRECT METHOD

The direct method (DM) simply adds the digital color's XYZ value to the background color's XYZ value to obtain the color blend. This is similar to the existing work by Weiland et al. [27]. Where they capture the background color using a High dynamic range camera placed on top of a head mounted display (HMD), then subtract this value from the sRGB to be displayed on the HMD. The display then shows subtracted results. They completely ignore both f_{render} and $f_{material}$ functions. Such that they take the sRGB colors as shown in Figure 9-A as the foreground colors for all the three OSTD displays. The algorithm 2 explains the direct method, where the prediction adds the given foreground and background as such, without accounting for f_{render} function .

Algorithm 2 Direct method based prediction algorithm

```
procedure DM-Prediction(Foreground,Background)
    Prediction=addXYZ(Foreground,Background)
return Prediction
end procedure
```

4.3 CAT METHODS

Chromatic Adaptation Transformation (CAT) are well established methods to estimate the colors a display can render based on the brightest white it can produce [25, 16]. Chromatic adaptation is the ability of a single color value to look different under different white lights for a human observer. CAT essentially is a transformation matrix that can transform a color based on the brightest white point (lighting condition). CAT has the potential to account for the f_{render} distortion function. Researchers have proposed CAT methods which rely on different matrices, as each matrix has worked for a certain set of color data. Three popular CAT methods chosen based on their popularity in the literature are: Bradford,Von Kries [25], and XYZ Scaling [16]. The CAT methods transform the display's color using the respective CAT matrix before adding to the background color. On applying the XYZ points given in table 1 to one of the three CAT method gives a CAT transformational matrix (Bradford,Von Kries and XYZ Scaling matrices). The resultant CAT matrices for all three OSTDs can be found in the Appendix . These CAT matrices are then applied on the XYZ value of the desired foreground color's XYZ to obtain the CAT transformed foreground color. All Formulas for the calculation CAT transformed matrices and CAT transformed colors are based on the formula given in the work by Susstrunk et al. [25]. The colors transformation according to the CAT matrices for all three displays are shown in figure 10.

Algorithm 3 takes one of the three CAT matrix for that particular display as a input parameter along with foreground and the background. Then this matrix transforms the XYZ value of the foreground to account for the f_{render} distortion function. The color blend prediction is done by adding this CAT transformed color value with the background.

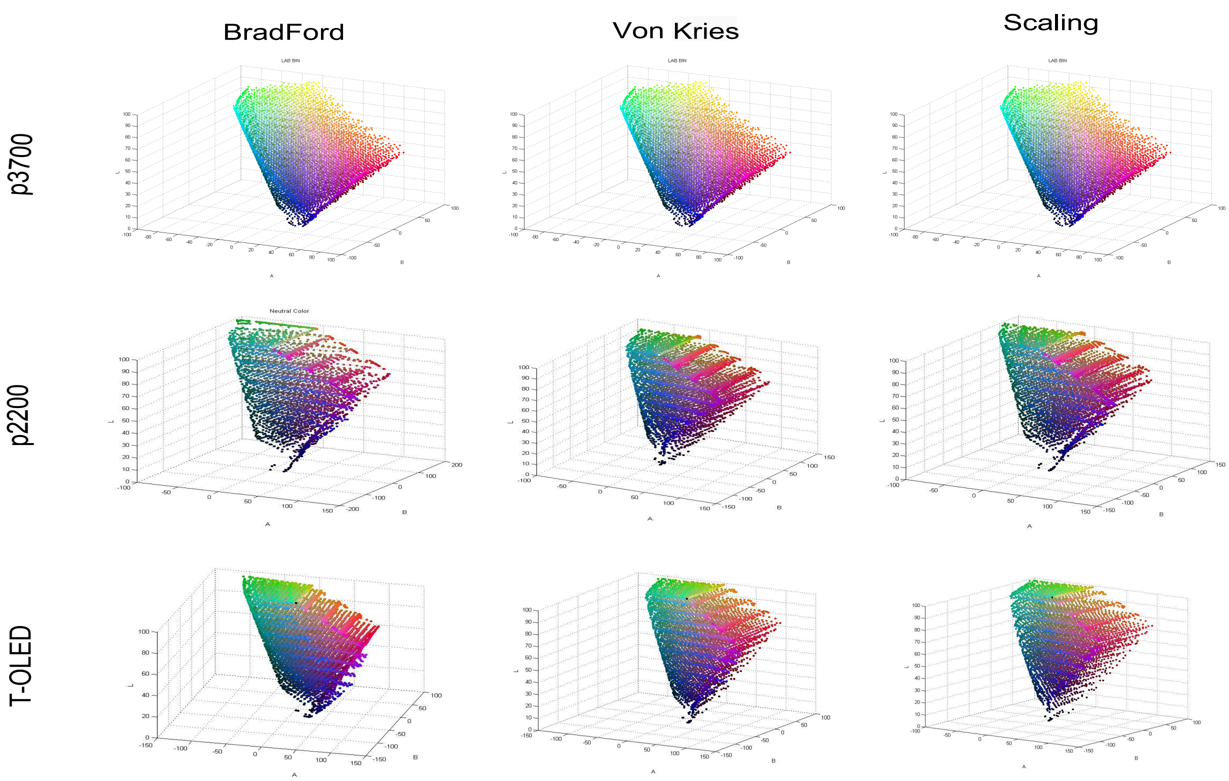


Figure 10: (Image best seen in color)CAT transformed foreground colors in LAB space for p3700, p2200 and T-OLED displays.

Algorithm 3 CAT method based prediction algorithm

---

procedure CAT-PREDICTION(*CATMatrix*,*Foreground*,*Background*)
  $CAT_foreground = Foreground \times CATmatrix$
  $Prediction{=}addXYZ(CAT_foreground, Background)$
return*Prediction*
end procedure

4.4 PREDICTION VALIDATION

The validity of BP method in the color blending model was tested by measuring the error of the color predicted and comparing it to the predictions errors in direct method (DM) and three chromatic adaptation transformation(CAT) methods. Prediction Error is the difference between the color predicted by the method and the measure of the actual color blend for the given set of foreground and background.

The difference is computed in CIE LAB. The colorimeter measured the color blend. This validation test compared prediction using all five methods on all three OSTD.

As discussed before, measuring the background color and characterizing the effect of the *material* distortion ($f_{material}$ function) is out of the scope of this work. The assumption of this work is that such color is available at a per-pixel level. However, to explore the impact of the *material* distortion, background detection was of two possible types : *plain* and *adjusted*. The background color is *plain* if the system ignores the effect of the *material* distortion, so in this case $f_{material}$(background color) = background color. The background color is *adjusted* if the system accounts for the *material* distortion and transforms it before feeding it to the model.

Background color consisted of 23 colors of the ColorChecker color rendition chart [19] at D65, ColorChecker is a representative set of naturally occurring colors. The background LCD monitor displayed all these 23 colors while the colorimeter measured them directly, these values correspond to the *plain* background condition as shown in Figure 8A. The colorimeter also measured the background color as seen through the OSTDs , these values correspond to the *adjusted* background configuration for each display as shown in Figure 8B-C. Foreground colors consisted of 838 random colors (10% of the size of the total bin).

4.4.1 *Data Collection*

Data collection process measured the resulting blend of each pair of foreground and background. Total measurement captured for a single display was of $23 \times 838 = 19,274$. So the total measurements for all three displays was $19,274 \times 3 = 57,822$. The colorimeter captured these blends in CIE XYZ, before being transformed into CIE LAB using the white points from Table 1. The resulting color blend for same set of colors was predicted for combination of display color method (5 methods), background configuration (2 configurations) and display (3 displays). Total data collected was $5 \times 2 = 10$ predictions per blending, $5 \times 2 \times 23 \times 838 = 192,740$ predictions per display, summing up to a total of $192,740 \times 3 = 578,220$. The prediction error was the Euclidean distance in LAB color space between each prediction and the actual measurement.

4.4.2 *Result*

Given the amount of data collected, data exploration needed a new visualization. Figure 11 shows the prediction results for a 838 random sample set on single random background color, on the p3700 display, with the *plain* background configuration, using the direct model. Figure 11A shows the prediction accuracy as a $3D$ shape in LAB space with more accurate predictions in dark red and less accurate ones in light yellow; the location of the points corresponds to the profile of the display. This $3D$ figure is instrumental in understanding which

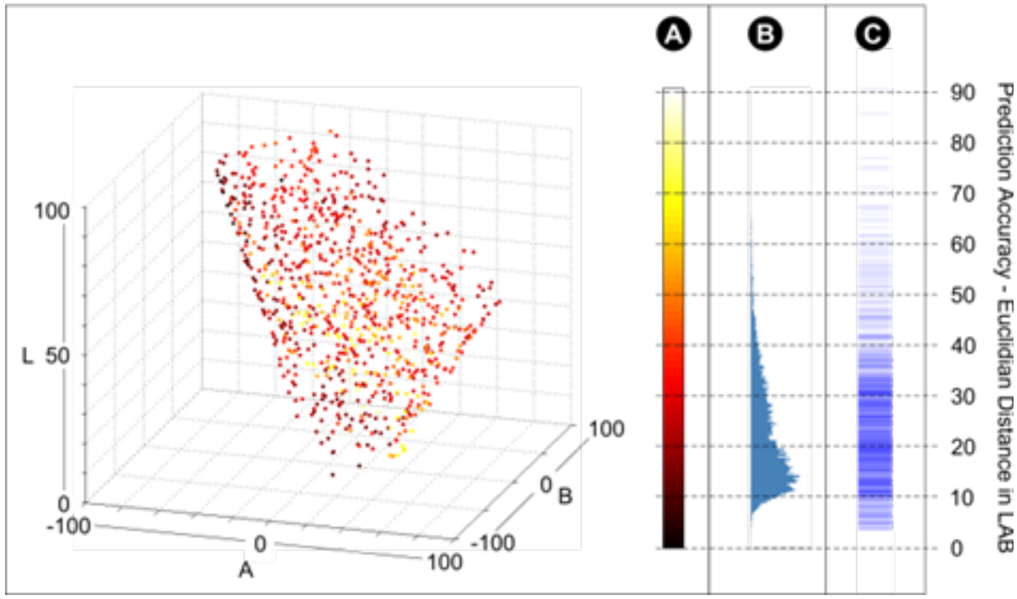


Figure 11: (Image best seen in color)Prediction result for 838 random foreground colors on a single random background in p3700 display. A) Accuracy per color in LAB; B-C) Histograms of the accuracy for the whole sample.

color areas are better predicted than others. Figure 11B shows a histogram of the same data points sorted by accuracy. More accurate predictions are piled up at the bottom near to zero, while less accurate predictions spread to the top. Figure 11C is a top view of this histogram with zero close to the bottom of the graph and color intensity representing the height of the histogram. The results analysis for prediction error uses this vertical histograms (new visualization).

Figures 12 and 13 make use of the vertical histogram to summarizes the results for our prediction for each type of display over five different methods and two different background types. Each histogram represents the prediction error, lower error or zero difference in LAB is in the bottom of the graph and color saturation from light to dark blue represents the height of the histogram. Dark blue represents the region where highest number of error values fell.

A visual inspection of the results shows that for all conditions the CAT-based predictions performed worst. This poor performance was expected as figure 10 showed that color transformed by CAT came no where near to actual color shown (figure 9). This result shows that CAT did not account for f_{render} as it was hoped to. All three CAT methods had a high spread in error and an average far from optimal. In the case of the p3700 display, all CAT methods perform the same because the white point of this display is equal D65. Thus the CAT methods is excluded from the rest of this analysis. The figures 12 and 13 also show that Binned profile method's the prediction results were the most accurate among all the methods both in plain and adjusted configurations. However in adjusted configuration the prediction accuracy was much more accurate than that in plain. The results for direct method also had a wide spread in prediction accuracy, this was true for both plain and adjusted configuration. The results for DM and BP shows that accounting just for render distortion sharply increase the prediction accuracy. Accounting just for material distortion does not increase the prediction accuracy. However on accounting for both render and material distortion increases the prediction accuracy.

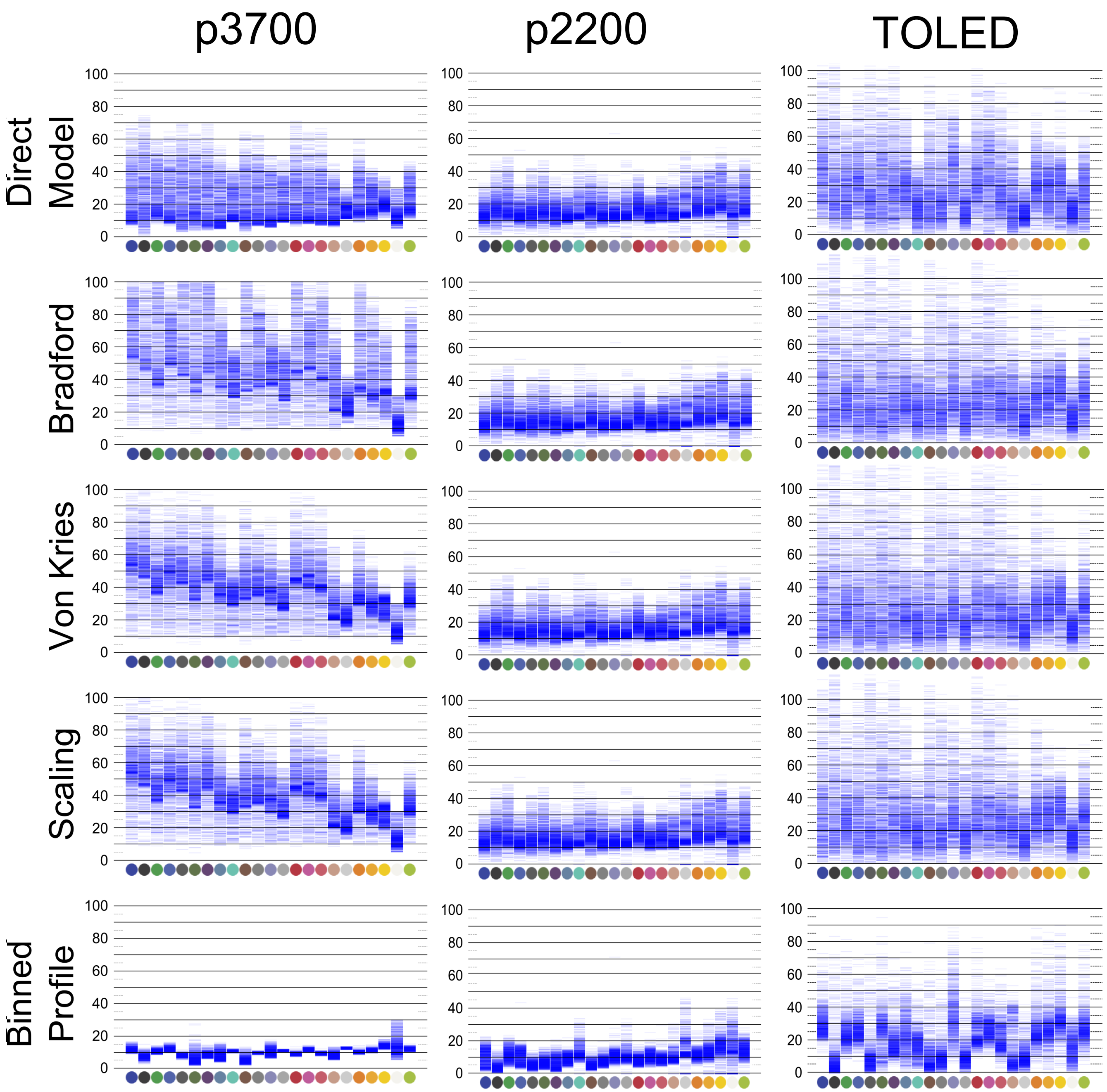


Figure 12: (Image best seen in color)Prediction results of p2200, p3700 and T-OLED displays, with 5 prediction methods, in *plain* bg configurations. With X- axis representing the background colors used.

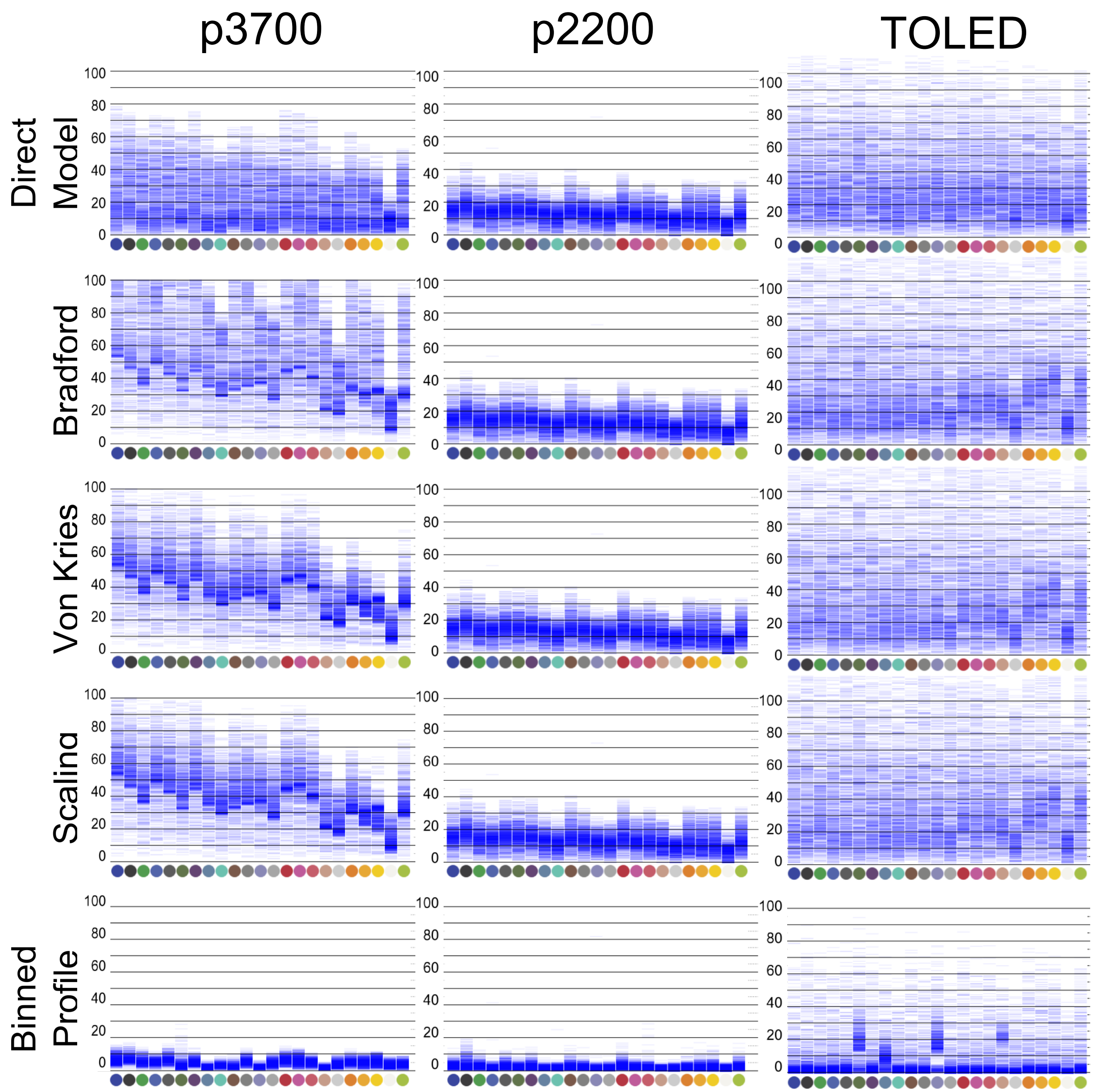


Figure 13: (Image best seen in color)Prediction results of p2200, p3700 and T-OLED displays, with 5 prediction methods, in *adjusted* bg configurations. With X- axis representing the background colors used.

Results did not have a normal distribution, this required the use of Kruskal-Wallis H test for non-parametric data. Kruskal-Wallis H test is a non-parametric method for testing if samples originate from the same distribution. This test is the non parametric equilateral of the one-way analysis of variance (ANOVA). It was done on the data collected to show how significant the difference where between the various main effects. Sig value less than 0.05 confirms a significant difference between the given effects.

Table 2 shows the results of our analysis. Results showed a main effect of *display, method* and *material* configuration. There were also significant interaction effects between all independent variables. On analysis of each display, for p2200 display BP-method performed best in both background configuration (*plain* : 10.01 - *adjusted* : 4.98). DM-method presented only a small difference between background configurations (*plain* : 22.71 - *adjusted* : 22.06). A similar pattern was observed for the p3700 display where BP-method has lower error for both background configurations (*plain* : 10.28 - *adjusted* : 2.77) than the DM-based prediction (*plain* : 17.5 - *adjusted* : 13.67). Finally, when applied to the T-OLED display BP-based prediction also performed better (*plain* :25.63 - *adjusted* : 8.24) than DM-based prediction (*plain* : 34.37 - *adjusted* : 32.26).

Quantitative analysis in Figure 14 shows that BP-method outperform all other methods. Moreover, this lower error rate exists for both the *plain* and *adjusted* background configurations. The quantitative also confirm the importance of the *render* distortion as the dominant factor for color blending, this can be seen in Figure 14 (difference in pink bars), where the error drops nearly by half on just accounting

Table 2: Kruskal-Wallis test for prediction error.

Display	Method	Bg-Type	df	χ^2	Sig
X	–	–	2	32152	<0.001
–	X	–	1	698210	< 0.001
–	–	X	1	25745	< 0.001
–	X	X	3	104717	< 0.001
X	X	–	5	101643	< 0.001
X	–	X	5	60583	< 0.001
X	X	X	11	142259	< 0.001
		Post	Hoc		
–	Both	Adjusted	1	59437	< 0.001
–	Both	Plain	1	21613	< 0.001
–	BP	Both	1	52494	< 0.001
–	DM	Both	1	2157	< 0.001

for *render* distortion in each of the 3 displays. The data shows that addressing *material* distortion, without accounting for *render* distortion does not produce significant change in the accuracy. However when *render* distortion is accounted for, addressing for *material* distortion helps to improve the accuracy. The results also very clearly highlight the limitation of the direct method and the inadequacy of any of the three CAT methods for color blend prediction. For the p3700 display prediction error using the BP method with the *adjusted* background was 2.77 or about 1 JND.

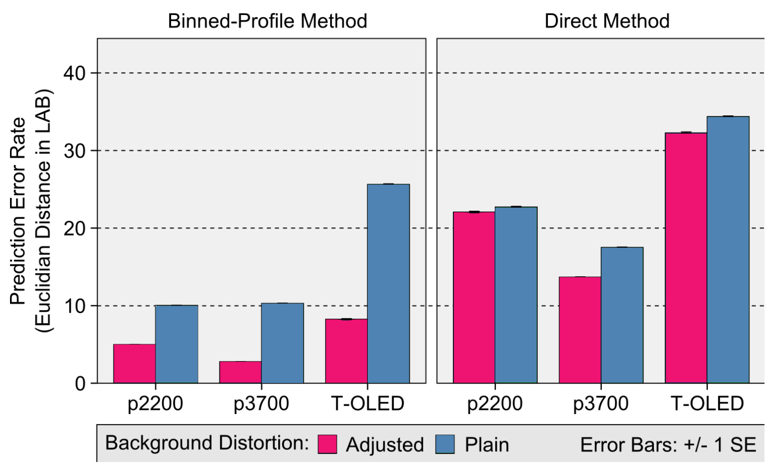


Figure 14: (Image best seen in color)Prediction error for the three displays, with the BP and DM , for two background configurations.

5

BP COLOR CORRECTION

Color correction aims at finding an alternative foreground color which, upon blending with the background, produces a color closer to the one originally desired by the designer. This section explains a color correction approach for optical see-through displays based on the BP method. At heart of color correction algorithm is the system's ability to predict how a particular foreground background color pair will blend. This prediction capability allows the system to find an alternate foreground color which comes closest to the originally desired color, for a given background. This is a best fit approach.

The BP Color Correction aims at correcting the color the display can actually show (i.e. color which has effect of *render* distortion), rather than the application defined foreground. This research work also avoids using color subtraction (CorrectedColor = foreground-background) as used by Weiland et al. [27]. for two reasons: first, as seen in the direct model for color prediction, direct color subtraction ignores the display profile which will lead to an incorrect target color for correction. Second, color subtraction often results in values which are outside of the display's color gamut [27].

5.1 BP COLOR CORRECTION ALGORITHM

BP Color Correction algorithm is described in Algorithm 4: There are three input parameters to the algorithm; Display type - which denotes which OSTD was used, Foreground - the sRGB color the system wants to paint on the screen, Background - the measured value of the background as seen through the OSTD (see Figure 8B). The sequential steps in the algorithm are as follows: *First*, the foreground's sRGB color is mapped to the closest bin in the LAB binned space. Then, based on the display profile, the bin color is mapped to its actual representation (DisplayForeground) of how that colors is shown by the given display. *Second*, for each representative bin color on the display profile(BP) the system predicts (Prediction) how it blends with the given background. This prediction is done based on color blending model given in equation 1.2. *Third* , the Euclidean distance in CIE LAB color space is calculated between the prediction and the display color (TempError). The system selects the color in the bin (ColorToShow) whose prediction had the lowest error/Euclidean distance. *Lastly* This color (ColorToShow) is then converted to its corresponding binned color via a reverse lookup to find the sRGB (CorrectedColor). Finally the display shows this corrected color. The flow of these steps are clearly shown in figure 15

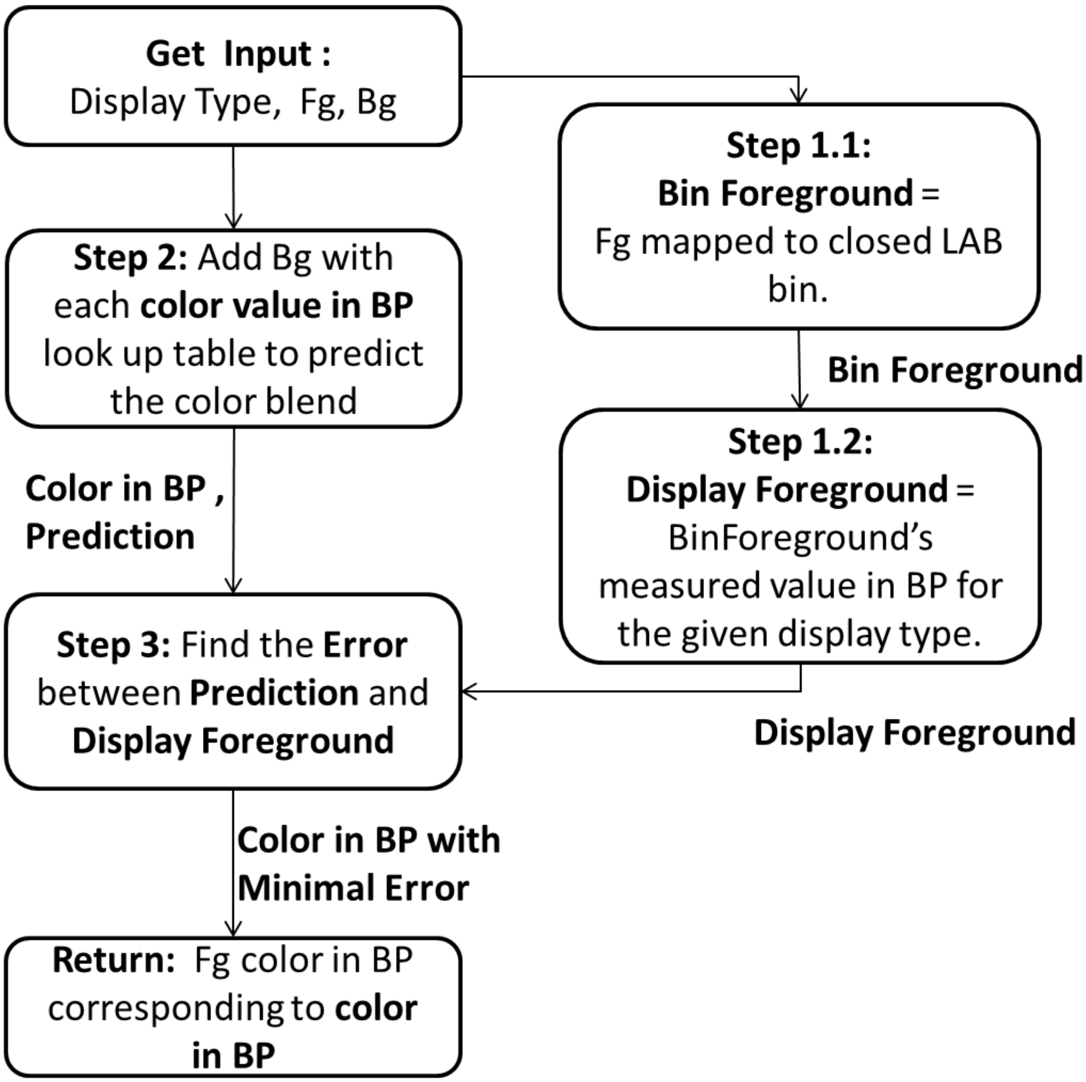


Figure 15: Logical flow of BP color correction algorithm

Algorithm 4 Binned-Profile color correction algorithm.

```
procedure BP-PRESERVATION (Display,Foreground, Background)
    BinForeground = findBin(Foreground)
    DisplayForeground = lookup(Display, BinForeground)
    Error = INFINITY
    for each Color in Display
        Prediction = addXYZ(Color, Background)
        TempError=distance(Prediction,DispForeground)
        if TempError<Error
            Error = TempError
            ColorToShow = Color
    CorrectedColor=revLookup(Display,ColorToShow)
    end for
    returnCorrectedColor
end procedure
```

5.2 DATA COLLECTION

With the accuracy of the BP method already proven, the primary focus of this study was not to compare how BP color correction performed with any other methods. Rather it is to explore how well the BP-based correction algorithm performs for varying set of foreground and background pair on various OSTD. This study applied the BP color correction on the p3700, p2200 and T-OLED OSTDs for the 23 ColorCheck's *adjusted* backgrounds. This study also used 200 random foreground colors. On BP color correction for a foreground color an alternate color is chosen by the algorithm, then this alternate color was shown each of the 23 backgrounds and to measured the resulting blend. The total measured data amounted to $23 \times 200 = 4600$ measures per display. For all three displays the total measurement was $23 \times 200 \times 3 = 13,800$. Correction error was calculated as the difference between the measured blend and the desired display color. The test bed was same as the one used in the color model validation.

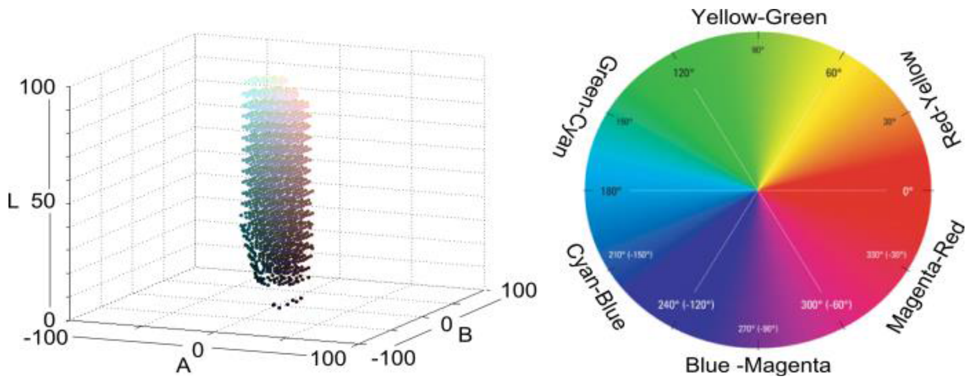


Figure 16: (Image best seen in color) Display color groups. Left: neutral colors within 10 JNDs from the L axis. Right: 6 chromatic regions - Yellow-Green, Green-Cyan, Cyan-Blue, Blue-Magenta, Magenta-Red and Red-Yellow.

To facilitate in-depth data analysis, The display colors were grouped into 10 subgroups: dark colors (L < 50), light colors (L >= 50), dark and light neutrals (neutrals are located within 10 JNDs of the L axis), and 6 chromatic regions according to the color circle. Figure 16 shows (left) the dark and light neutrals, and (right) the 6 chromatic regions. Note that each display color might belong to more than one group. Similarly, the ColorCheck backgrounds were divided into high intensity colors (L >= 50) resembling daylight conditions like white and yellows, and low intensity colors (L < 50) resembling night conditions like black and blue. Figure 17 shows the background color groups.

Figure 17: (Image best seen in color) Low intensity and High intensity background groups.

5.3 COLOR CORRECTION ANALYSIS

The analysis of collected data was done in two steps. The first analysis looked into the general correction capacity of the algorithm for the three displays. In the second one focus was only on the p3700 display, as its display capability had large color spread (see Figure 9C-E for the color profile of each display).

Analysis of the correction results was done using the vertical histogram. A color heat-map (see Figure 19-Top-Right) was also developed. The color heat-map reveals how well the algorithm corrects regions of the LAB color space for a given set of background and foreground colors. This color heat-map divides the LAB D65 slice into a 30×30 grid. Each grid cell is colored in blue (#0000FF) with the opacity moving from 0 to 1. The opacity of blue was relative to the average correction error (ranging from 0 to 100+) of all colors in that cell. If the sample did not contain corrections for display colors in a given cell, the cell has no blue box. In case more than one sample fell on the same grid , the error of each sample's correction is calculated and averaged with the rest. Cells in which colors have lower correction error are in faint blue hue. Cells in which colors cannot be corrected in average (higher correction error) result in a dark blue. Example of this heat map is given in the Figure 18.

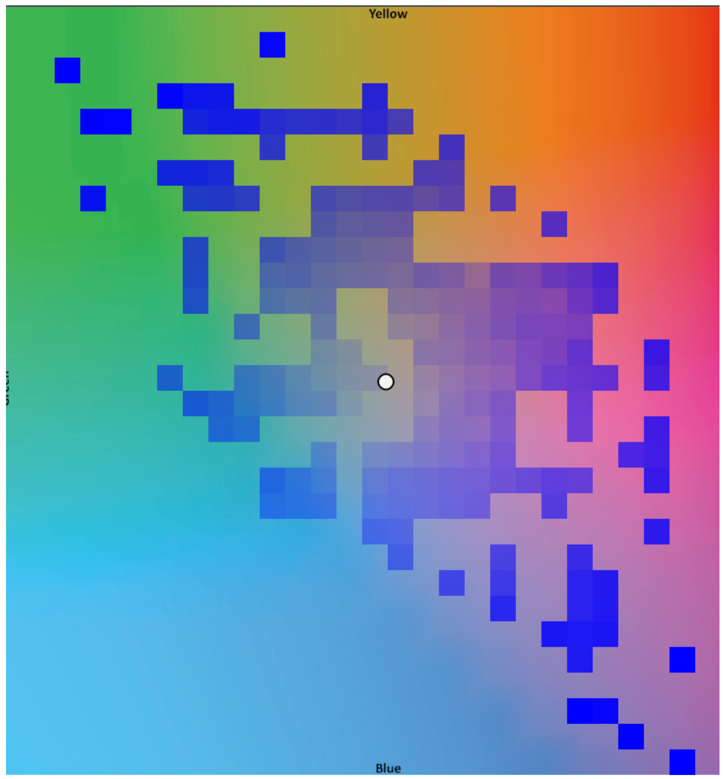


Figure 18: (Image best seen in color) Example of heat map with various colors for TOLED display and how they are corrected for the background color white. Dark blue hue represents colors which are less corrected, light blue representing colors which were corrected better.

5.3.1 *Display wise analysis*

Figure 19 shows the general correction results for all background and display colors on the three displays. The results reveal in all three displays the correction works better for low luminosity backgrounds (First 11 background color left to right in vertical histogram, as explained in figure 17) when compared to high luminosity background (Last 12 background colors). Results also show corrections have lower error for the p2200 and T-OLED displays, this can be seen by looking at the fainter blue boxes in the heat-map and more concentrated vertical histograms. This was due to the limited range of colors these displays render. As shown in figure 9 both these displays had a small concentrated volume in the LAB color space and therefore the distance between the measured correction and the target color will always be small.

Conversely, corrections have higher error for the p3700 display. p3700 in contrary to other two OSTD had a wider range of colors as seen in figure 17 and therefore the distance between the measured correction and the target is considerable larger. Finally, display colors toward the edge of the gamut (red, green, blue) generally had a higher error rate when compared to the colors located in the central region of the gamut. Most color spotted on the edge of the gamut were result of the prefect solution being outside the gamut of the display, this in turn contributes to the error rate as they are not the prefect solutions.

5.3.2 *Color Region analysis*

The aim of this analysis, was to have an understanding about the correction capacity of various digital color. The previous works on color correction both on projectors [22, 2, 20, 1] and on OSTD [27] have stated, that not all color can be corrected, as some of the solutions to color blending lies outside the color gamut of the

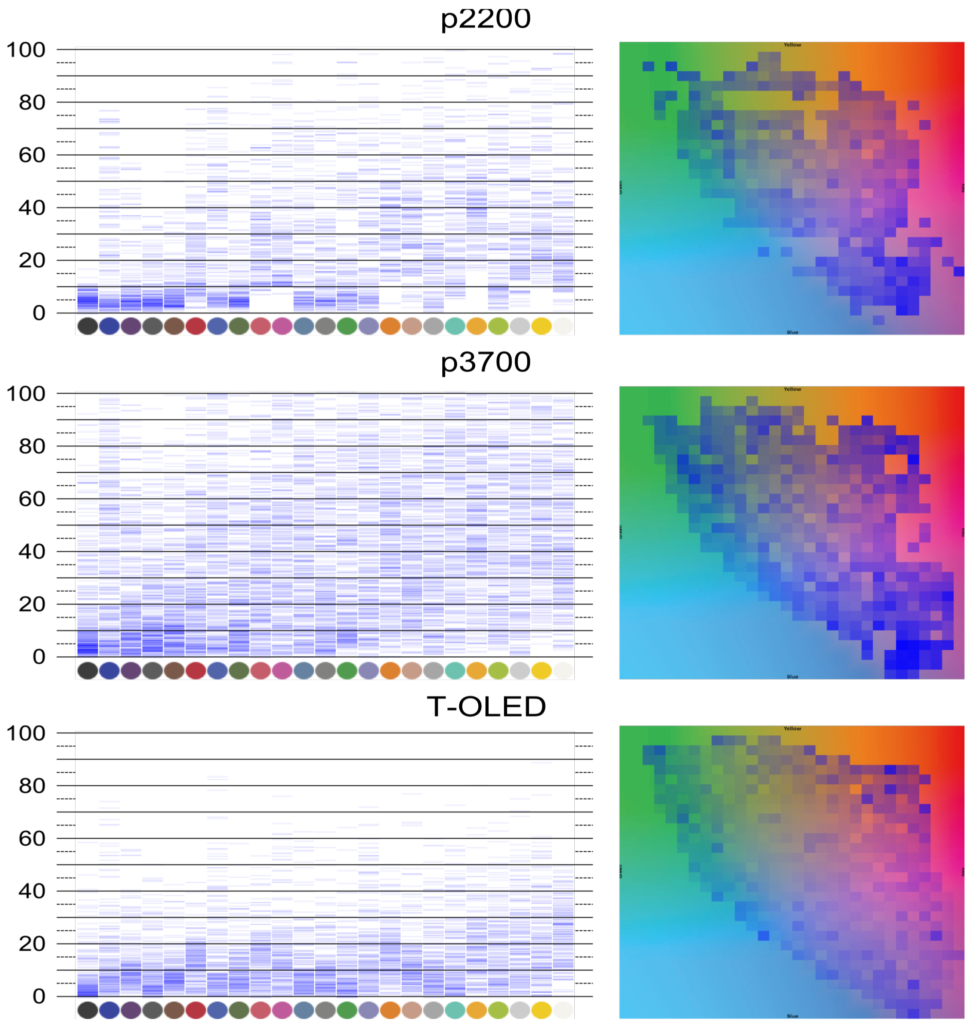


Figure 19: (Image best seen in color) Overview of correction error rate. Heatmap - darker blue indicates higher correction error.

display. However all these previous studies have failed to capture the correction capacity of the various digital color under varying background. This analysis was omitted in the color correction works in field of Spatial Augmented Reality [22, 2, 20, 1]. This in most cases was attributed to the fact that the background colors in these works were static. However in case of OSTD like HMD or Mobile displays the backgrounds will change. So it becomes imperative to understand which foreground colors will be effective on varying background condition.

The OSTD p3700 had the largest color profile of the 3 OSTD, as you would expect in a general purpose multimedia device. This fact made the color region alaysis in the p3700 display more meaningful than in other two OSTDs. So shown in section 5.2 the foreground color of the p3700 were separated into 10 subgroups. The background colors was also divided into two low and high Intensity, low intensity background emulates the colors present in dark environments or night conditions and high intensity backgrounds emulates the ones in daylight conditions. Figure 20 shows the correction results for the p3700 display according to high and low intensity backgrounds. The correction error heat map for light and dark backgrounds and light and dark neutrals are shown. The various coloured points in the Color space on top of the heat map indicates the background colors used.

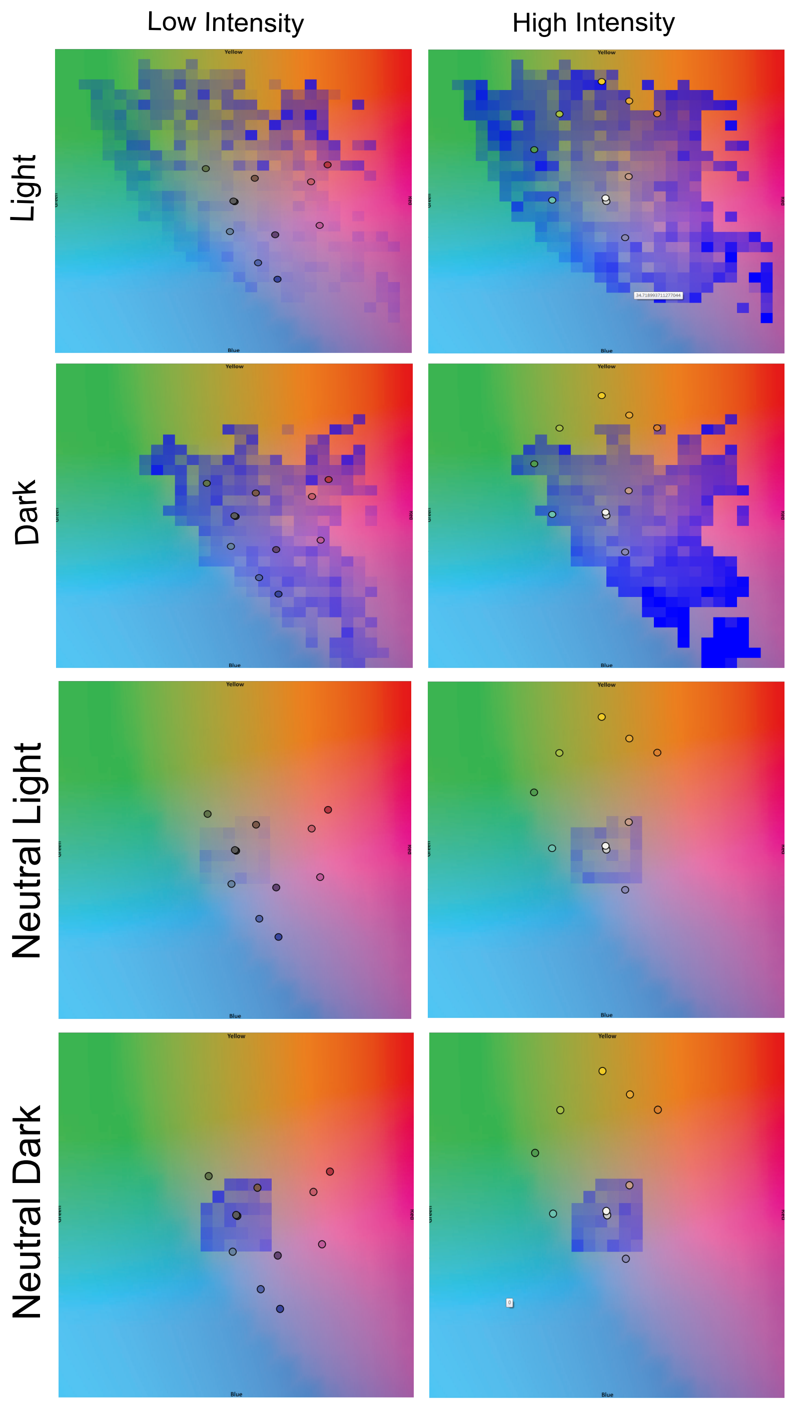


Figure 20: (Image best seen in color) Correction error heat map for p3700 with low and high intensity backgrounds represented as columns and foreground color conditions represented as rows.

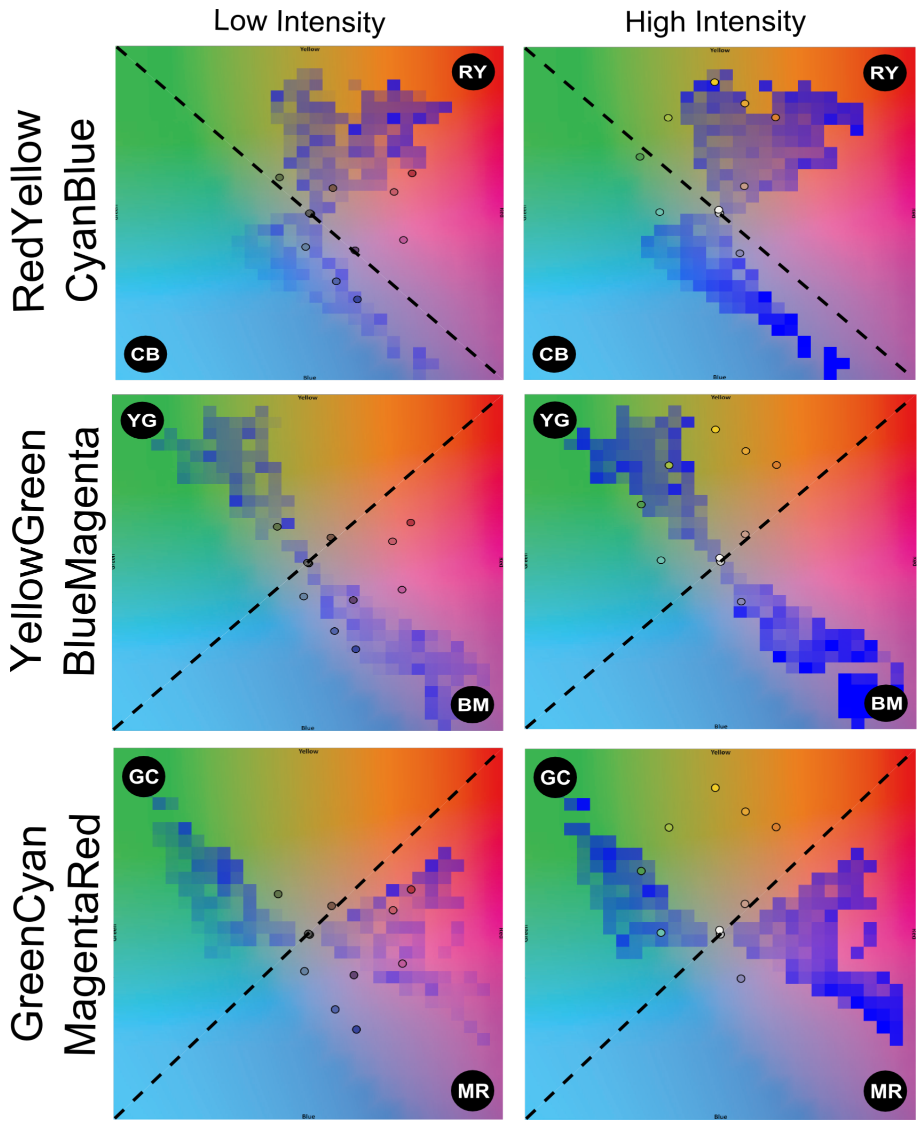


Figure 21: (Image best seen in color) darker blue indicates higher correction error. Foreground regions.YG:Yellow-Green, GC:Green-Cyan, CB:Cyan-Blue, BM:Blue-Magenta, MR:Magenta-Red and RY:Red-Yellow. Points the heat map represent backgrounds used.

A visual inspection of Figure 20 shows that BP-based color correction for the p3700 display works best on low intensity (dark) backgrounds. For high intensity (light) backgrounds a decreased correction capacity across all display colors was observed. The result shows a significant decrease of correction capacity for dark display colors. For both background conditions the light neutrals present lighter heat-maps. Correction for dark neutral colors were found to performing better than most the correction for the most dark foreground colors. It can be said that based on this result that the region of neutral colors might be larger than what was originally assumed, as similar levels of correction error can be found at a bigger radius around the neutral colors.

Table 3: Kruskal-Wallis test for correction error.

Region	BG	Display	Neut	df	χ^2	Sig.
All	–	–	–	5	23.9	< 0.001
–	Both	–	–	1	1056.5	< 0.001
–	–	Both	–	1	761.9	< 0.001
–	–	–	Both	1	46.5	< 0.001
		Post	Hoc			
–	Low	Both	–	1	381.7	< 0.001
–	High	Both	–	1	651.1	< 0.001
–	Both	Dark	–	1	684.2	< 0.001
–	Both	Light	–	1	685.4	< 0.001
–	Low	–	Both	1	6.3	< 0.05
–	High	–	Both	1	66	< 0.001
All	Both	Both	–	23	1979	< 0.001

Figure 21 shows the heat maps of correction error for the 6 chromatic regions on the p3700 display. Almost all color groups had very high correction error under high intensity background. For low intensity backgrounds Cyan-Blue, Blue-Magenta and Magenta-Red are corrected best, however, for high intensity backgrounds it is Red-Yellow, Yellow-Green, Green-Cyan and the neutrals that are corrected best. This can be seen also from the figure 22 which shows the quantitative analysis of the color regions for each of the two the background intensity. The data for the correction error was not normally distributed and therefore Kruskal-Wallis H test (see Table 3) was used to analyse the data. The results show there is a significant difference between corrections in all chromatic regions, between both background intensities, and between both display color luminosity (light and dark color) conditions ($p < 0.001$). Post-hoc tests show there was a significant difference between all possible combinations of conditions (all $p < 0.05$). Results also show all colors are better corrected in low intensity backgrounds. However, neutral colors are always corrected significantly better than non-neutrals. In general corrections had lower error for low intensity backgrounds at 21.23 (9.2 JNDs), for light foregrounds at 23.46 (10 JNDs), and the CyanBlue region at 31.49 (13.6 JNDs).

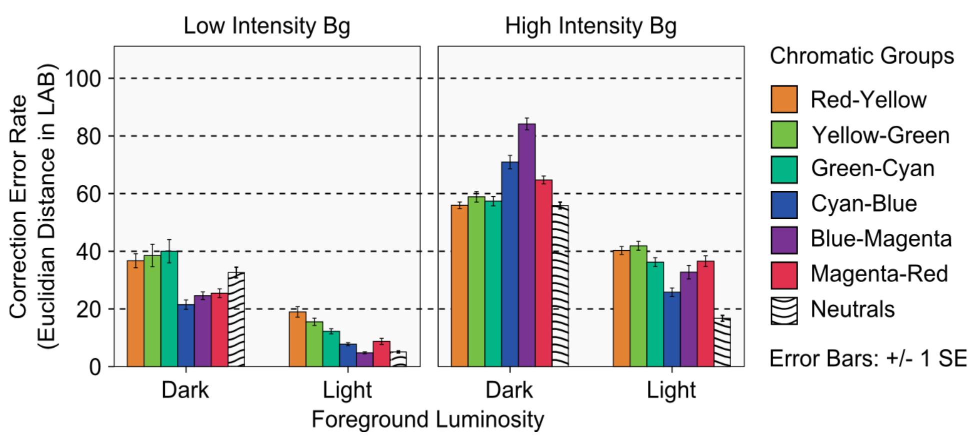


Figure 22: (Image best seen in color) Quantitative analysis of correction error using the BP model for the p3700 display.

5.4 comparison with trivial compensation

As discussed before in section 2.2 Weiland et al. [27] developed a color correction based on RGB subtraction. RGB subtraction based color correction in Weiland et al. [27] is called as "Trivial Compensation". Trivial compensation involves directly subtracting the RGB color of the digital foreground with the background's RGB color captured by a camera. This work by Weiland et al. [27] is the state of the art for color correction in OSTDs. In this section the comparison is made between Weiland's Trivial Compensation and BP method of color correction.

The comparison is done in two different ways. In the first comparison, ten thousand random set of color blends are corrected both using trivial compensation and BP color correction. The resultant hue value of the corrected color on blending with the background is compared with the hue value of the intended foreground. In the second comparison sample images are corrected on various background using both trivial compensation and BP color correction. The results are presented for viewing.

5.4.1 *Random Color Blend Comparison*

In this comparison ten thousand random foreground and background are chosen in RGB , their blend is corrected both using trivial compensation and BP color correction. In trivial compensation simply the foreground is subtracted from the background as explained by Weiland et al. [27]. Then the resultant color is added with the background to find the corrected color. This color addition was done in XYZ color space. The hue value of the corrected color is calculated. The same was done to BP color correction as well. However in case of BP, the background color was first converted in XYZ and steps given in section 5.1 were followed. The final corrected color is then taken and its XYZ value was added to the background's XYZ value to find the final corrected color's hue. The figure 23 shows the comparison graph between the final hue on correction of trivial compensation and BP color correction.

It can be seen from the graphs in figure 23, the spread of hue for trivial compensation is very high and random, while BP color correction has a more linear graph. This shows that the colors chosen by BP color correction as solution tend to be of similar hue to that of the desired foreground.

High spread in trivial compensation can be attributed to the fact that, on using RGB subtraction in foreground background combination where the background color's RGB value is greater than foreground's RGB , the resulting corrected RGB tends to zero. This was also clearly explained in Weiland et al. [27] work as the major disadvantage. In these graphs for 9024 foreground colors out of 10000 where either R or G or B value is not equal to zero, trivial compensation solution had R or G or B value equal to zero respectively. In comparison BP color correction created only 4671 such colors. For trivial compensation 9024 color amounts to 90% of all colors in compared, while for BP color correction it was only 40%. It is to be noted as stated earlier in this chapter not all colors can be corrected by BP correction, this might be a potential reason behind slight discrepancies in hue shown in BP color correction.

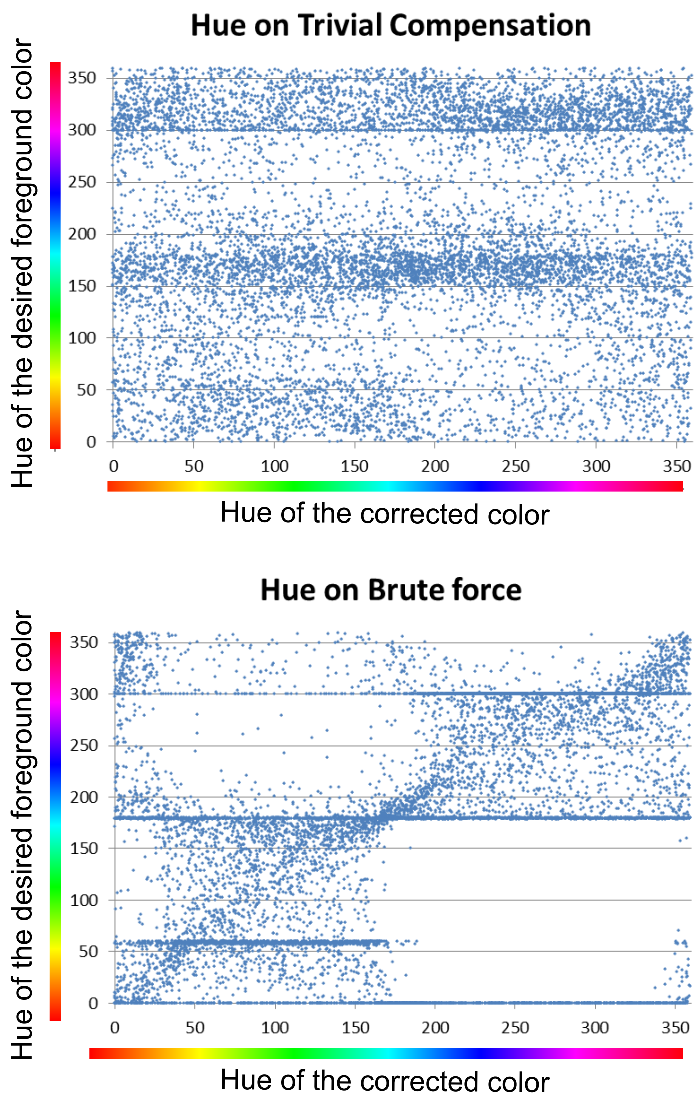


Figure 23: (Image best seen in color) Graph showing hue comparison for 10000 random colors with, X-axis representing hue of the color on correction upon blending with the background and Y-axis as the color of the desired foreground color. The graph on the top represents the trivial compensation's results and one in the bottom represents the BP color correction results

5.4.2 *Digital Image Comparison*

This section shows the performance of BP color correction with trivial compensation on set of digital images taken with the context of everyday use. The aim here was to show how well BP correction can preserve the colors and a comparison with trivial compensation. The corrections were run on a simulation which takes in a background and foreground, calculates how these two colors will blend using CIE XYZ addition, then corrects the foreground image based on both trivial compensation and BP color correction receptively. The comparison between the blend, trivial compensation and BP color correction are shown in the figure 24. Here it can be clearly seen how BP color correction performs in preserving the color. There has been noticeable improvement from the color blended image and the BP color corrected image. The trivial compensation on the other hand performed noticeably more poor than the BP color correction.

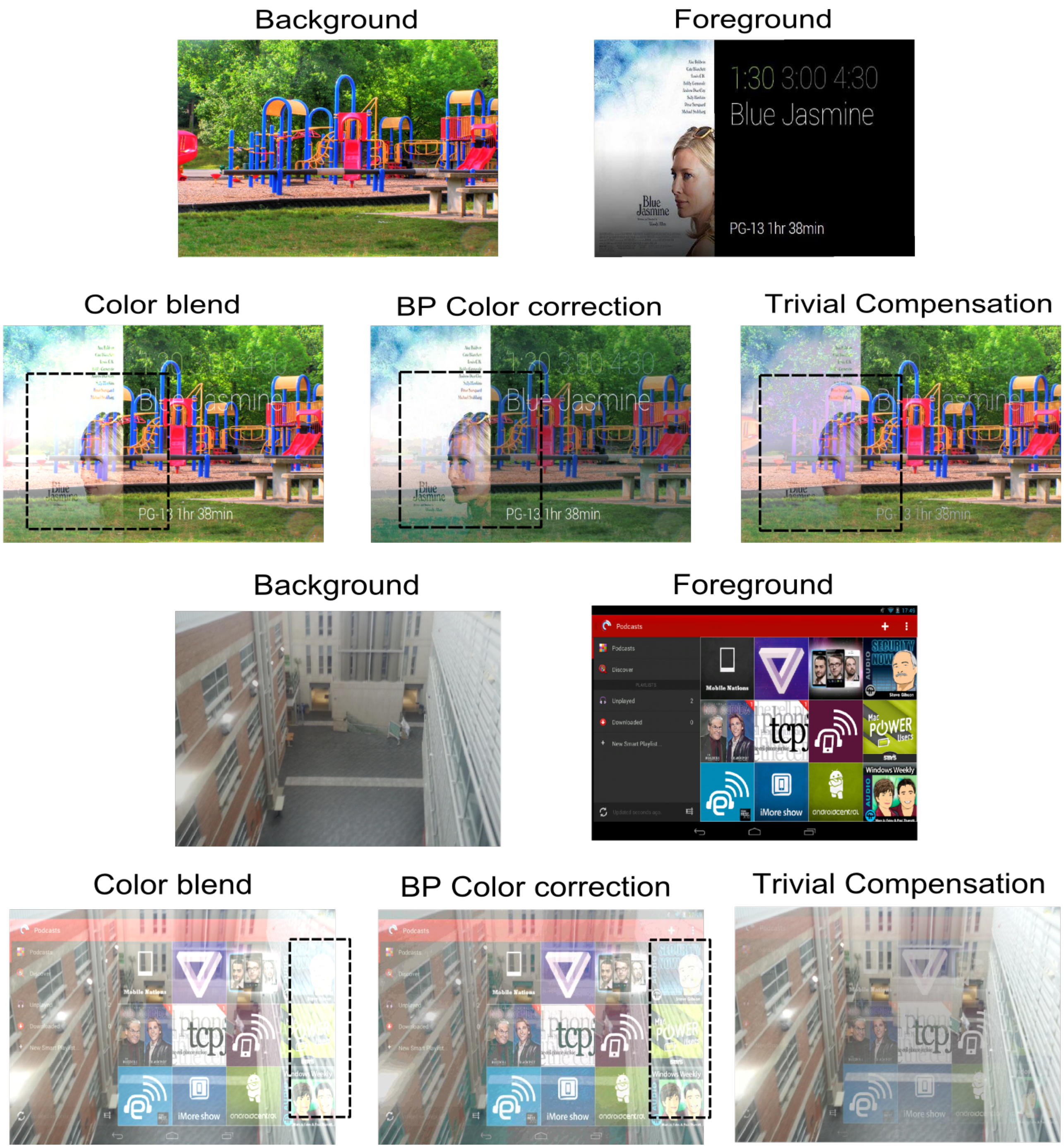


Figure 24: (Image best seen in color) Comparison of BP color correction with trivial compensation on every day background with standard head mounted display interfaces in a simulation. Dotted black box highlights the corrected region

As explained in previous section for a high intensity background the trivial compensation pushes the foreground colors to black when foreground's RGB value is less than background's RGB value. The example of for this is given in figure 24 (top). This results in the final blended image upon trivial compensation being barely visible. This fact was also explained by Weiland et al. [27]. However BP color correction corrects the colors considerably better. In case of low intensity background as seen in figure 24(bottom) trivial correction does not tend to zero, however the corrected hue value are very different from that of the original foreground image. BP color correction on the other hand corrects the foreground color and looks close to the original foreground image.

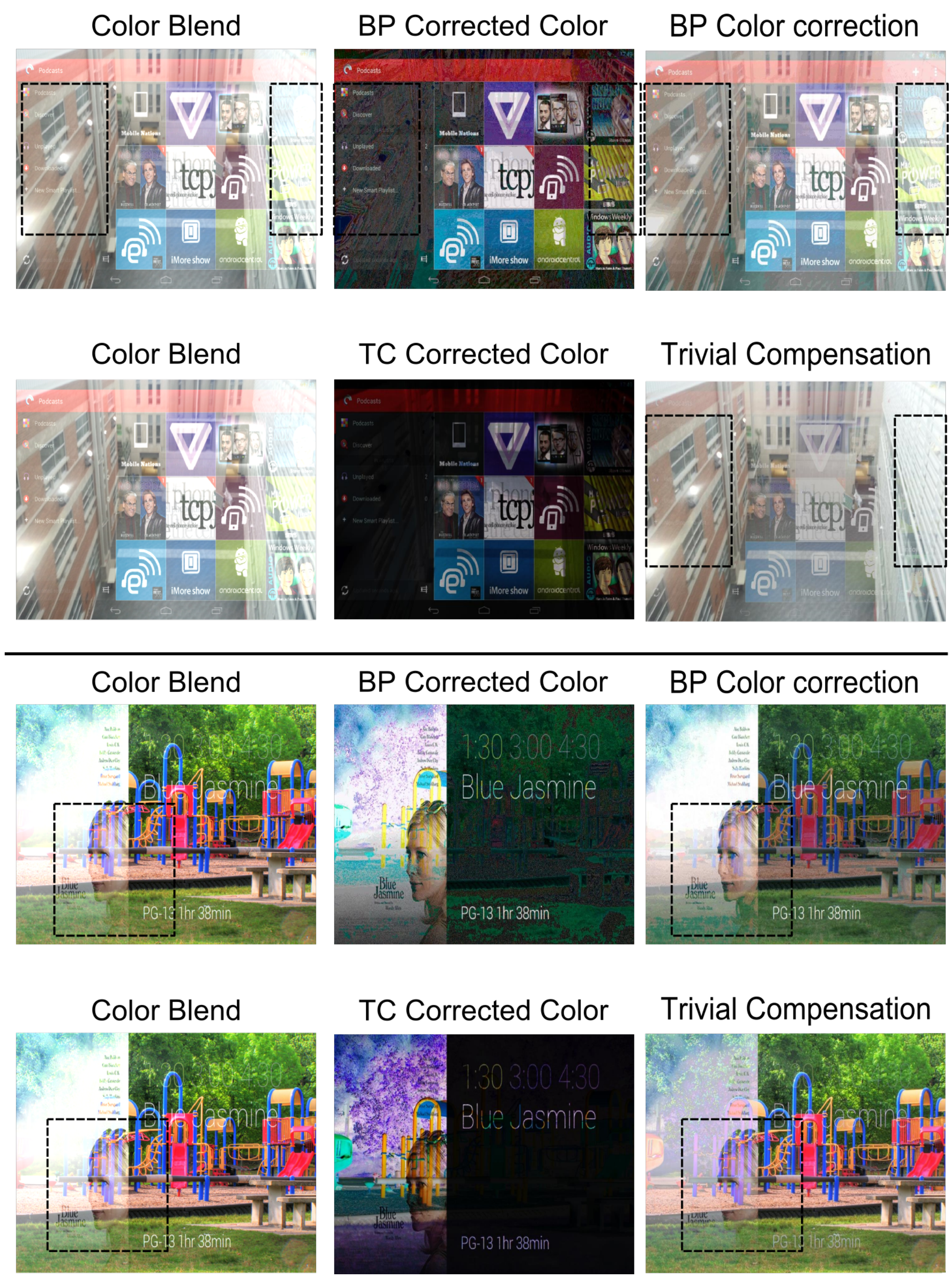


Figure 25: (Image best seen in color) Comparison of BP color correction with trivial compensation. Top: Correction on a high intensity background, Bottom: Correction on a low intensity background. The Image in the middle shows the output of the correction algorithms. Note that the top image's TC (trivial compensation) corrected colors are mostly back, where the background is of high intensity.Dotted black box highlights the corrected region.

6

QUICK CORRECTION ALGORITHM

The BP color correction algorithm is a brute force solution for the color blending problem in OSTD, it looks at every possible color in the binned profile (8390) before picking the best one. When used at a per pixel level this approach results in a large calculation over head. OSTD in general has rich multimedia content with large pixel resolution, in such a display a real time algorithm for color correction is required for the solution to be effective. This chapter explores one of the possible approaches to have a color correction algorithm which corrects at a faster rate.

6.1 SCOPE

This sections aims to develop an approximate color correction algorithm which does a limited number of calculations (<8390). The binned profile has an organised data structure of colors in LAB space ($5X5X5$), however this data structure was not used in BP correction to traverse towards the solutions. A new algorithm called quick correction was developed to traverse the BP profile data without looking into every possible color. Quick Correction uses directional searches in the BP data set, where the color to compare starts at the center of the BP data set and moves in the direction of the best possible solution. This section will explain quick correction algorithm and will compare it with BP color correction, to establish its usability. The larger picture of this sections is to set a course for future works with respect to development of real-time solutions for color blending.

6.1.1 *Quick Correction*

CIE LAB is a continuous color space, bin profile is a small sub set of this space. Quick correction extends LAB 5X5X5 binning entire color space, the color correction for any foreground color start, at the mid point of this space (50,0,0) and moves in the direction of the approximate solution to the color blending. This section will explain in detail how quick correction algorithm works.

Algorithm 5 gives the steps involved in quick correction. This algorithm takes the display type, foreground and background color as input parameters. Quick correction starts at the center (Origin) of the binned profile which has its initial value set to 50,0,0 in LAB Bin. The correction error (OriginError) at the origin is calculated. This is the distance between the foreground color and the corrected color blend with the background. Error is low when the distance between the blend and the foreground is low and error is high when the distance between the blend and the foreground is high. Six sample colors called "Samples" (top, bottom, front, back, left and right) at a distance of "StepSize" are selected around the "Origin". "StepSize" is initially set to max values of LAB (LAxisMaxValue,AAxisMaxValue,BAxisMaxValue) which represents the boundary of the LAB color space (with L:1 to 100,A:-100 to 100 and

B:-110 to 110). The correction error of each the selected samples (SampleErrors) is then calculated . If OriginError is found to be lower than the all six samples errors, then the StepSize value is reduced by half ("StepSize"/2). Iteratively the algorithm is repeated till some samples have lower error than the "Origin". When some or all "SampleErrors" were found to have lower error than the "OriginError", weights for each sample are calculated based on their error value.The weights are only assigned to samples which have correction error lower than the Origin. Highest weight is given to the sample with lowest error. Lowest weight is given to the sample which has the highest error. Displacement is calculated based on these weights. A new origin is selected (NewOrigin) by applying this displacement on the current origin. If the NewOrigin is outside the BP (empty) then the StepSize value is reduced by half ("StepSize"/2). Iteratively the algorithm is repeated until the NewOrigin value is not empty. Correction error for newly selected origin is calculated. If the new origin has higher correction error than the "Origin", then the StepSize is again reduced by half ("StepSize"/2). Iteratively the algorithm is repeated until the NewOrigin has lower error than the "Origin". If the NewOrigin has lower error than the "Origin" then the "NewOrigin" is set as "Origin". The algorithm runs Iteratively till the "stepseize" reaches zero. When the step size reaches zero, the final Origin will have a color with lowest correction error. This Origin's color is returned as the output.

Figure 26 illustrates how weight calculation and displacement are applied to select a new origin. Figure 27 shows an example condition under which the step size is reduced.

Algorithm 5 Quick correction algorithm.

```
procedure Quick correction(Display,Foreground,Background)
    BinForeground = findBin(Foreground)
    DisplayForeground = lookup(Display, BinForeground)

    Origin=Center of Binned Profile
    Blend = addXYZ(OriginColor, Background)
    OriginError = distance(Blend, DisplayForeground)

    StepSize = 3DPoint(LAxisMaxValue, AAxisMaxValue, BAxisMaxValue)
    SamplesColors[6]=ColorsAroundOrigin(top, bottom, left, right, front, back)

    While StepSize is Not (0,0,0)
        for each Color in SamplesColors
            Prediction = addXYZ(Color, Background)
            SampleError[index]=distance(Prediction,DispForeground)
            if SampleError[index]< OriginError
                Samples Closer Than Origin ++
        end for

        if SamplesCloserThanOrigin == 0
            Reduce StepSize by half
            Continue
        else
            Calculate Weight for samples accurate than Origin
            Calculate Displacement based on Weights

        NewOrigin = Origin + Displacement

        While NewOrigin is empty
            Reduce Displacement by half
            NewOrigin = Origin + Displacement
        end While

        Blend = addXYZ(NewOrigin, Background)
        NewOriginError = distance(Blend, DispForeground)

        ifOrigin == NewOrigin
            Reduce StepSize By Half
        else ifOriginError <= NewOriginError
            Reduce StepSize by half
        else
            Origin = NewOrigin
    end While
    CorrectedColor=revLookup(Display,Origin)
    returnCorrectedColor
end procedure
```

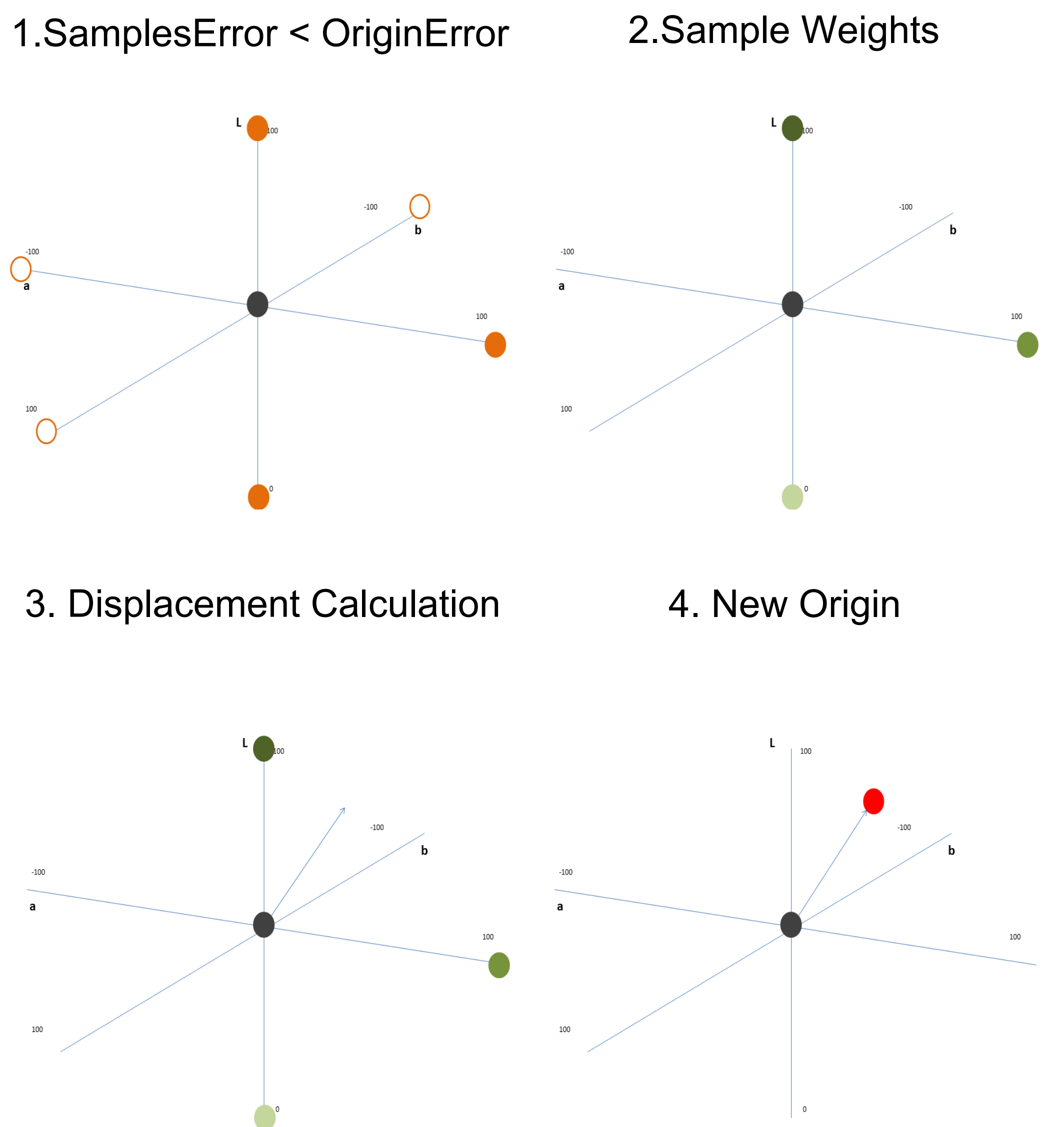


Figure 26: (Image best seen in color)Example image explains the weight and the displacement, 1: Three orange filled circles represent the samples with correction errors lower than Origin, 2: Samples are weighted according to their error , dark green - higher weight (low error), light green lower weight (high error), 3: Calculation of displacement from the current origin and 4: New origin selected based on the displacement(red)

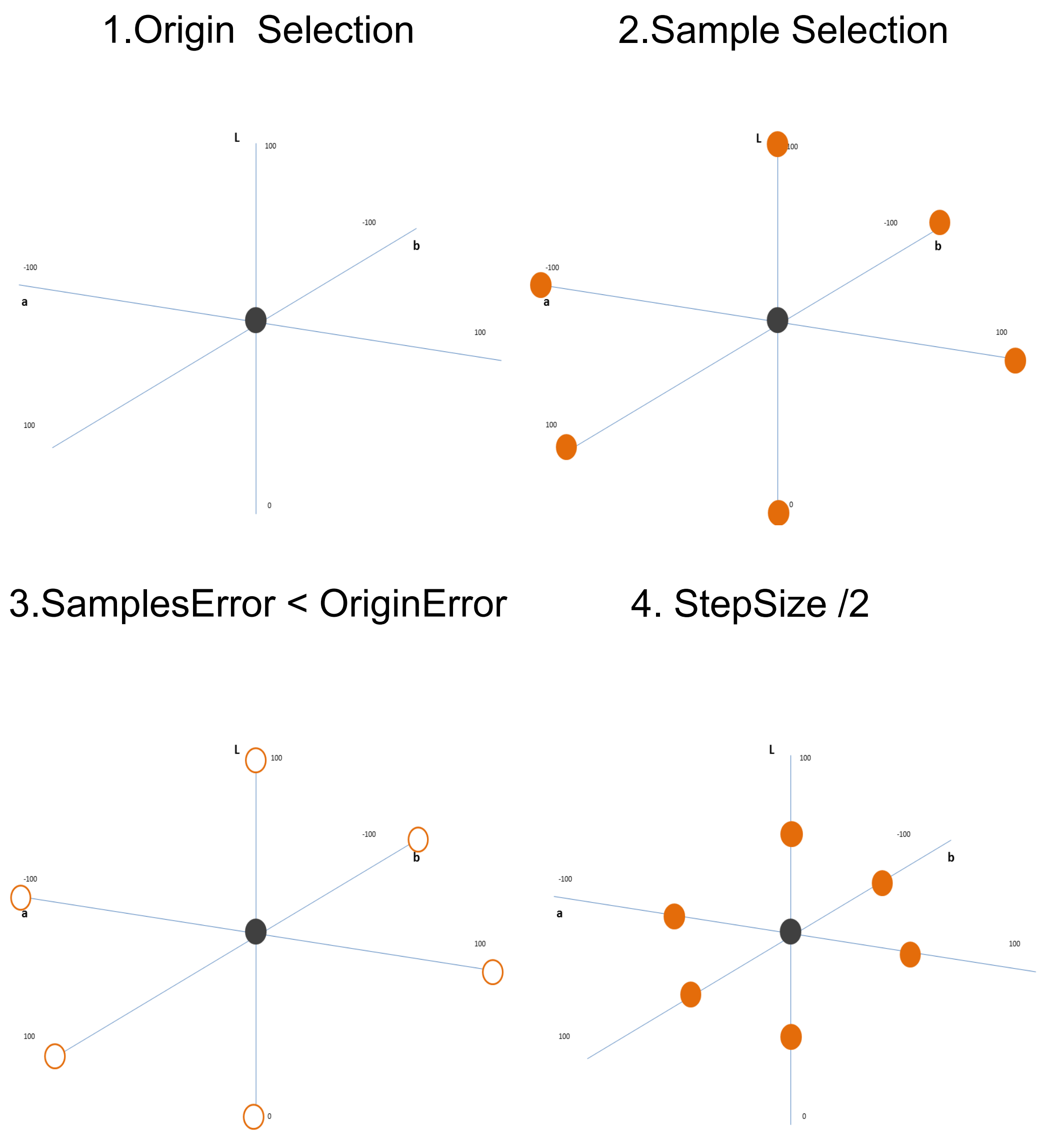


Figure 27: (Image best seen in color)Example image explains the Step Size Reduction, 1: Black circle represents the initial origin selected , 2: Orange filled circles represents the six selected samples around the origin, 3: Empty orange circles represents the samples which have correction error greater than the origin and 4: shows the step size reduction by half

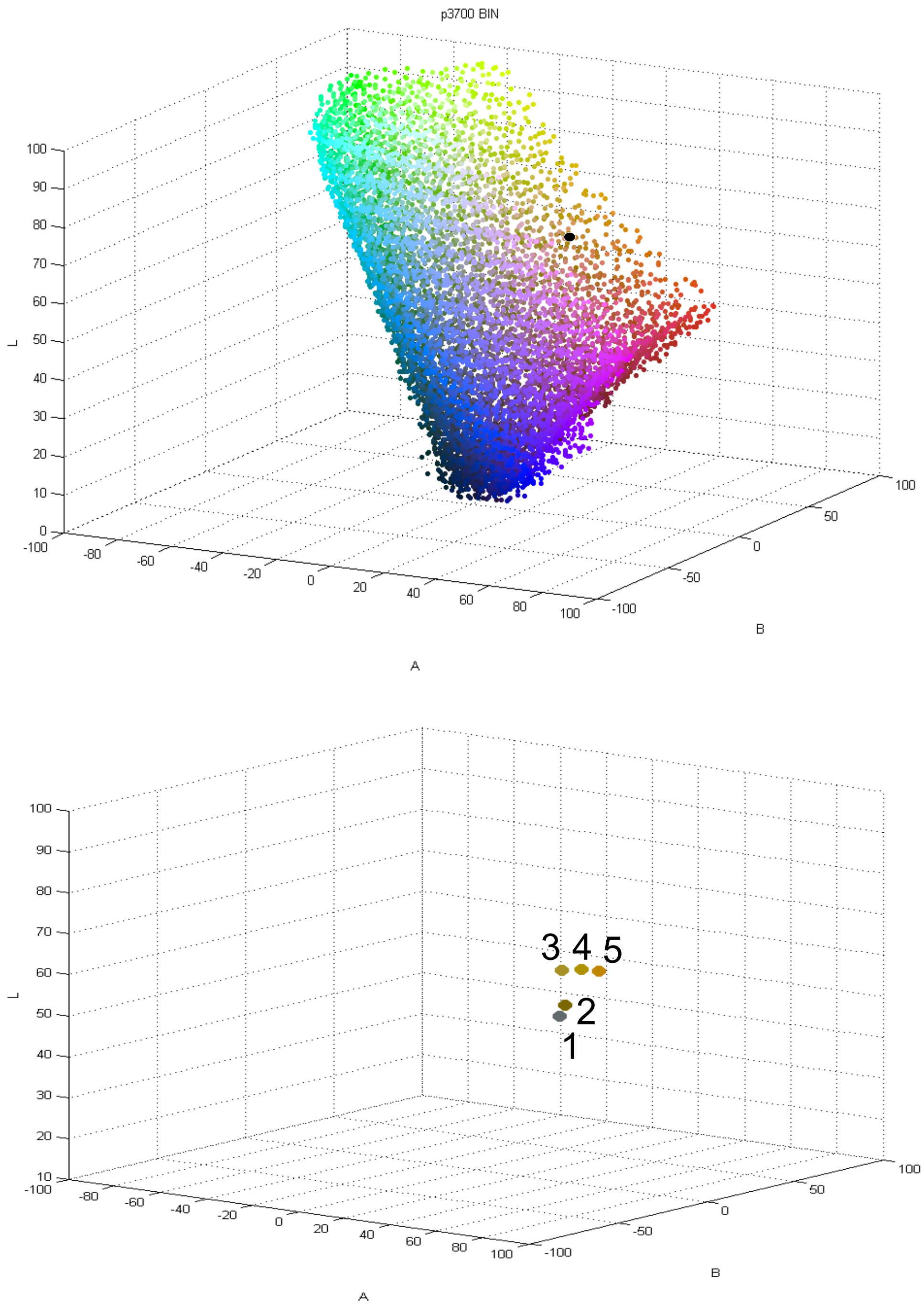


Figure 28: (Image best seen in color)Top, p3700 BP with , BP color correction result masked as black for yellow foreground ($FFFF00$) and gray(side walk) background. Bottom, Quick correction's directional jumps moving towards the solution

Figure 28 shows how quick correction jumps to a new points (NewOrigin) on p3700 BP, for foreground color yellow and background color gray (Side-walk). It can be seen that corrections starts from origin (marked as 1 in the figure 28) and moves in the direction of the solution (marked as 5 in the figure 28). Figure 28 also shows how the quick correction jumps in comparison with solution picked by BP corrections for the same foreground background condition. The result selected by BP is marked as black. It is to be noted that figures 28 hides the steps which reduces the step size by half, as in those cases the origins will remain the same. Unlike BP color correction the quick correction is not designed to pick the most optimal solution, rather it picks a color closer to optimal within a short time. The optimal solution can be missed by quick correction as the algorithm stops when the step size is zero.

6.2 comparison with bp correction

As explained in section 6.1.1 quick correction (QC) was not designed to pick the most optimal solution to achieve color correction, rather to pick an approximate solution in a short time span. This section looks how approximate the solution picked by quick correction is when compared to BP color correction. This comparison is done in two steps. In the first comparison ten thousand random foreground background combinations were chosen and corrected using both BP color correction and quick correction. The second comparison is between the digital images corrected using both BP color correction and quick correction, to showcase the visual difference.

6.2.1 *Random Color Comparison*

BP color correction as shown in chapter 5 always picked a color which comes closest to the intended foreground. Quick correction solution's accuracy was tested using the BP correction's solution as the ground truth. To establish how close QC's solution is to BP correction, two comparisons were made: the hue comparison and the distance comparison. As seen in BP correction analysis (section 5.3), there are several factors behind a color's correction capacity such as intensity of the background color, foreground color's luminosity and color regions. To find how the accuracy of the QC solution varied, all these factors were recorded in this comparison.

In hue comparison, the hue difference between the color chosen as the solution by the BP correction and quick correction was calculated for all ten thousand foreground background combinations. The results can be seen in figure 29. An average of 7°(in a hue angle scale of 0-360°) change was noticed for light foregrounds in low luminosity background, this was the highest average change among all the foreground background combinations. The lowest change in hue of 3°was seen in dark foreground for low intensity backgrounds. This shows the change in hue was very minimal showing that the QC solution indeed moves in the direction of BP correction chosen color.

In distance comparison, the euclidean distance was calculated between the color chosen by quick correction and the color chosen by BP corrections, both on blending with the background. CIE LAB space was used to find the color difference as per the formula given in chapter 1. The figure 30 shows the shows the quantitative difference measure for various foreground and background. The color difference of 16 (6 JND) between QC and BP was the highest recorded difference, it was for light foreground and high intensity background condition. The best case accuracy of 9 (4 JND) was found with darker foreground colors on low intensity backgrounds. The analysis was also done based on six color regions (see section 5.2), to understand how QC performs for various foreground colors. Figure 31 shows the distance between QC's and BP's chosen colors in blending for various foreground color region. The color in cyan-blue and blue-magenta region were closest chosen by QC when compared with BP, that too in low intensity background it was as close as 6 JNDs (3 JND). This can be related to the fact that BP had best correction in this region among all the other color regions (see section 5.3). Yellow-green region exhibited the maximum QC selection mismatch when compared with BP, particularly the yellow-green colors on hight intensity background had the worst result - 24 (11 JND). The results from hue comparison and distance comparison shows QC did pick a color of similar hue but the accuracy when compared to BP was lacking.

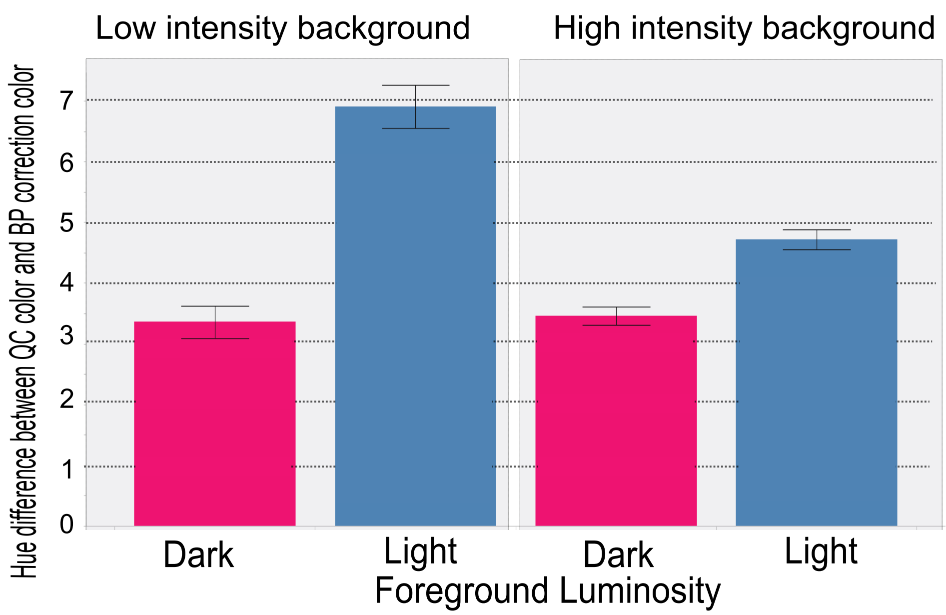


Figure 29: (Image best seen in color) Graph representing hue difference value between QC chosen color and BP chosen color value.

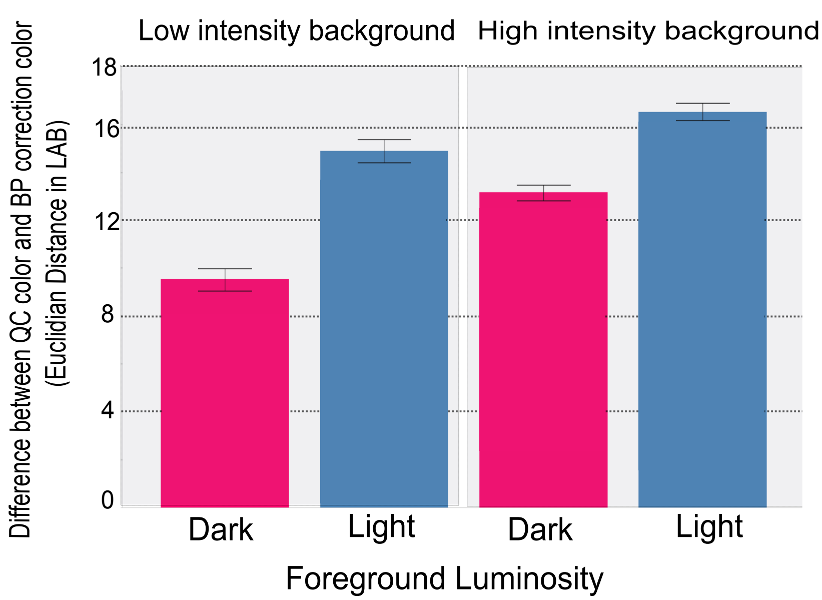


Figure 30: (Image best seen in color) Graph representing color difference value between QC chosen color and BP chosen color value.

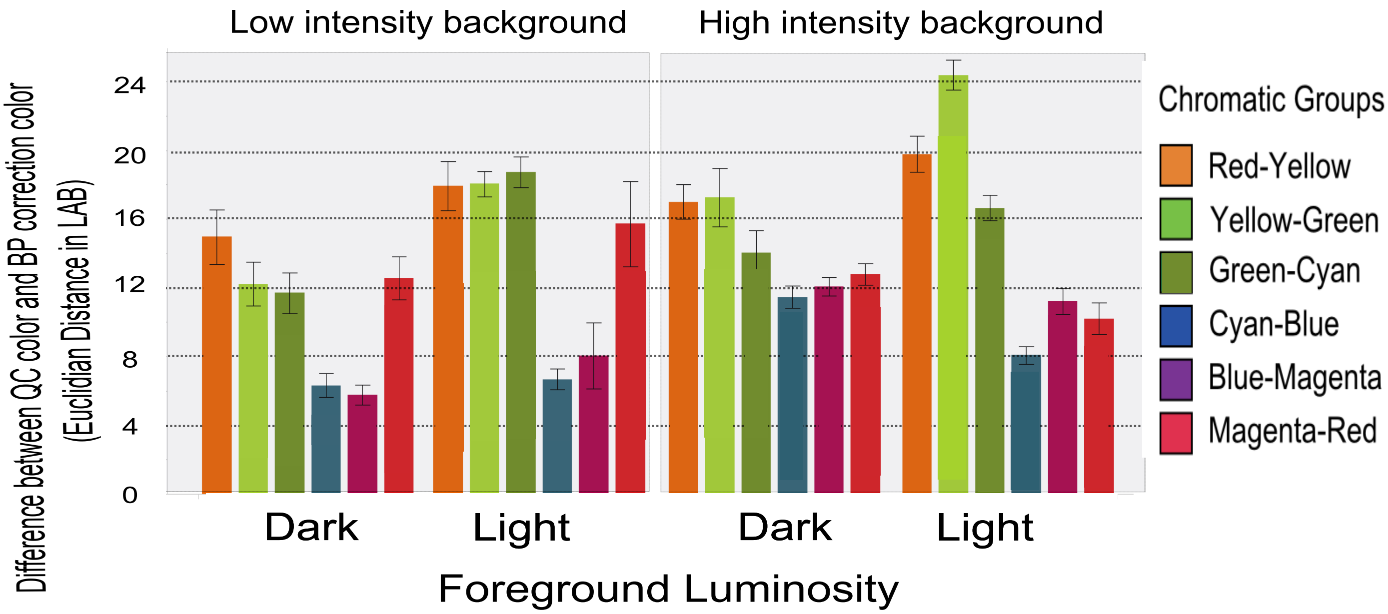


Figure 31: (Image best seen in color) Graph representing color difference value between QC chosen color and BP chosen color value with six color regions.

6.2.2 *Digital Image Comparison*

In this section various digital images shown in section 5.4.2 were again corrected using quick correction using the same simulation. Figure 32 shows the colors blends being solved with BP color correction and quick correction, as it can be visibly seen in most cases quick correction performs almost similar to BP color correction.

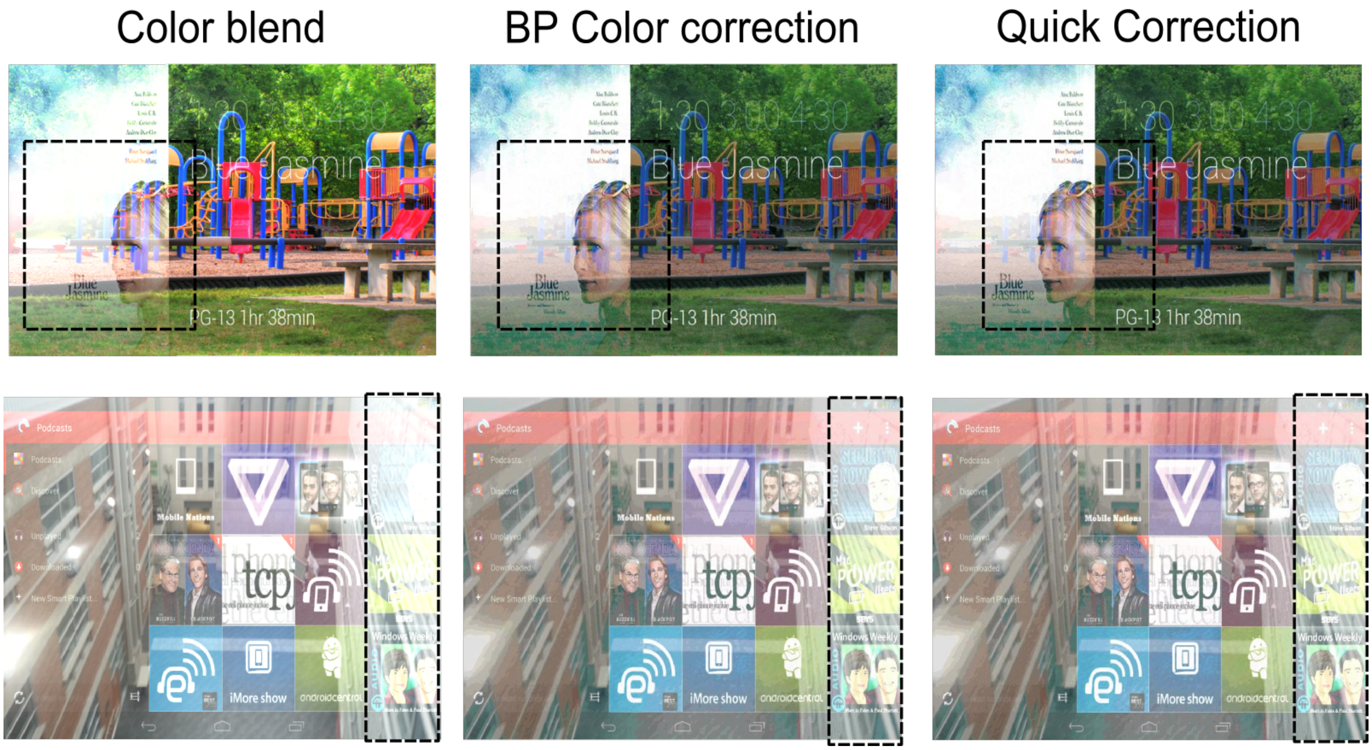


Figure 32: Comparison of quick correction with BP color correction on every day background with standard head mounted display interfaces in a simulation

6.3 PERFORMANCE

The aim behind the development of quick correction is to have a color correction algorithm which has a faster processing time to pick an approximate solution for color correction. Performance of BP color correction and quick correction was monitored on two different PC with different CPU and GPU capacity. The first PC's CPU hence forth called as CPU1 is Intel $i7-3770$, $3.40GHz$ eight core processor. Its GPU hence forth called as GPU1 is NVIDIA Quadro 600 with 96 CUDA cores. The second PC's CPU hence forth called as CPU2 is

Intel xeon $E5 - 1620.3.60GHz$ eight core processor. Its GPU hence forth called as GPU2 is NVIDIA Quadro 8000 with 256 CUDA cores.

Four different size of foreground image were put through pixel wise color correction upon a similar size background image. The four size chosen were $1280X800$, $1024X768$, $960X540$, $240X320$. These four sizes were chosen to represent maximum pixel resolution of p3700, p2200, a standard off the shelf HMD and T-OLED display respectively. The tables 4, 5 and 6 have the runtime for a single frame to be corrected on various processors and GPUs over three different profile sizes. The correction was done on a simulator. For GPU implementation CUDA programming was used. Improvement in execution time on average for a quick correction is 99% more faster than BP correction when compared on a similar GPU. Equation 6.1 shows the formula used to calculate the improvement. There is also visible decrease in the execution time for quick correction with decrease in the size of the BP.

$$Improvement = (\frac{BPExecutionTime - QCExecutionTime}{BPExecutionTime}) \times 100 \tag{6.1}$$

The overall results shows that quick correction can potentially run 99 times faster than BP color correction with minor accuracy error in the final solution. These results need to be tested further with a experimental set-up. This algorithm was developed with intention to give space for future research to look into in the needs and feasibility of real time correction.

Table 4: Run time comparison using p3700 BP .

Corrector	Processors	1280X800	1024X768	960X540	240X320.
BP	CPU1	176 days	135 days	89 days	13 days
	CPU2	157 days	121 days	79 days	11 days
	GPU1	4.92 minutes	4.44 minutes	3.46 minutes	7 seconds
	GPU2	3.39 minutes	2.26 minutes	2.15 minutes	5 seconds
QC	CPU	18.8 seconds	8 seconds	6 seconds	1 seconds
	CPU2	9.3 seconds	8.6 seconds	8.8 seconds	1 seconds
	GPU1	2.7 seconds	2.2 seconds	1.5 seconds	0.152 seconds
	GPU2	2.8 seconds	2.4 seconds	1.1 seconds	0.81 seconds

Table 5: Run time comparison using p2200 BP .

Corrector	Processors	1280X800	1024X768	960X540	240X320.
BP	CPU1	176 days	135 days	89 days	13 days
	CPU2	157 days	121 days	79 days	11 days
	GPU1	4.92 minutes	4.44 minutes	3.46 minutes	7 seconds
	GPU2	3.39 minutes	2.26 minutes	2.15 minutes	5 seconds
QC	CPU	8.7 seconds	8.4 seconds	6.4 seconds	1.1 seconds
	CPU2	8 seconds	7.5 seconds	8.9 seconds	1 seconds
	GPU1	1.5 seconds	1.5 seconds	1.5 seconds	0.116 seconds
	GPU2	2.3 seconds	2 seconds	1.1 seconds	0.68 seconds

Table 6: Run time comparison using TOLED BP .

Corrector	Processors	1280X800	1024X768	960X540	240X320.
BP	CPU1	176 days	135 days	89 days	13 days
	CPU2	157 days	121 days	79 days	11 days
	GPU1	4.92 minutes	4.44 minutes	3.46 minutes	7 seconds
	GPU2	3.39 minutes	2.26 minutes	2.15 minutes	5 seconds
QC	CPU	6 seconds	5.6 seconds	7 seconds	0.798 seconds
	CPU2	6.6 seconds	6.1 seconds	8.5 seconds	0.901 seconds
	GPU1	1.8 seconds	1.8 seconds	697	0.132 seconds
	GPU2	1.7 seconds	1.4 seconds	1.3 seconds	0.49 seconds

7

MATERIAL DISTORTION

The work in this thesis is established under the assumption that measurements of the background color are available accurately at all time. This section tries to understand the material distortion and possibly look into characterizing it for the different display material used in this thesis. Solving material distortion stays outside the scope of this work.

In most of the previous works on color correction such as Bimber et al. [2] and Weiland et al. [27] a camera was used as a color measuring instrument to capturer the background color . The calibration process required to achieve high precision measurement using a camera is a challenging problem and there exists a long line of research works which aims to achieve this [6][15][21]. Even with a calibrated camera, accurate measurement of color is still a complex task as explained by Hong et al. [8]. However given the recent research works in the field of digital image processing and its due research progress, a camera with high precision color measurement is achievable.

On availability of such high precision camera, they can be integrated with OSTD to capture the background color. The practicality of such an integration in a OSTD such as HMD demands the camera to be placed in a way that it does not disrupt the observer's line of sight. Weiland et al. [27] placed their camera on top of the HMD screen, in such a set up that the background color measured will always be in *plain* configuration. The chapter 4 section 4.4.2 shows and explains accounting for *material* distortion helps in improving accuracy. The practicality of measuring the background color creates the inherent need to characterize a relation between the *plain* and the *adjusted* configuration for a given OSTD. This section explores one possible approach to characterize this relation.

7.1 SET-UP

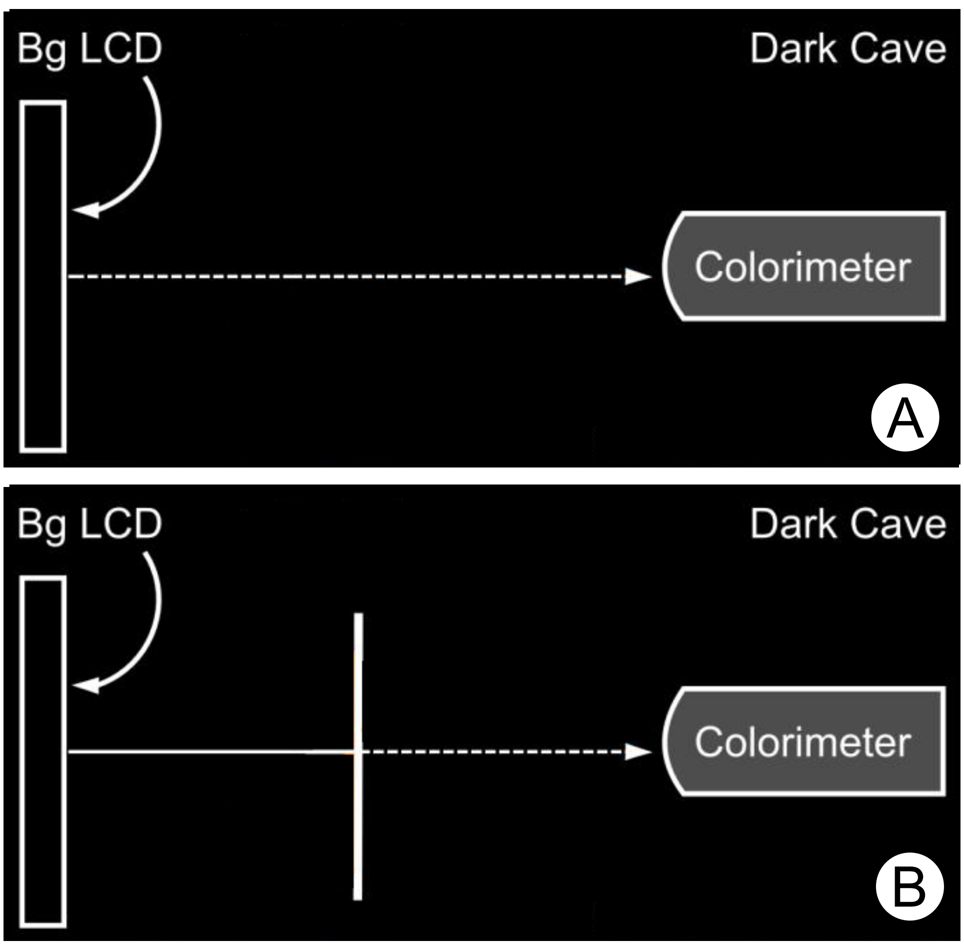


Figure 33: A-*Plain* configuration, B-*Adjusted* configuration

Set-up is shown in figure 33, both the Lumisty and T-OLED based OSTD were used. CS200 colorimeter was used to measure the displayed color value. Dell U2312HM VGA LCD display calibrated at the standard D65 white point was used to generate the colors. A total of 8390 colors were shown by the display, the colors were the same as the bin colors used in BP. These colors were chosen as they represent the whole of sRGB color space. All the displayed color were measured in two conditions, one without the OSTD (*plain* configuration) and one through the OSTD (*adjusted* configuration).

7.2 APPROACH

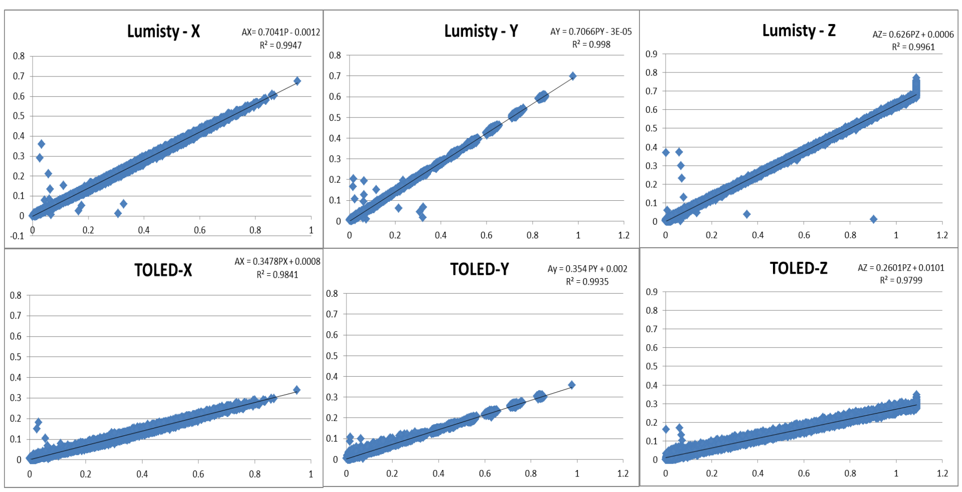


Figure 34: The linear relation between the *plain* (X-axis) and *adjusted* (Y-axis) configuration. Measurements are in CIE XYZ

The aim was to predict a color value in *adjusted* configuration from a color captured in *plain* for a given OSTD. Both *plain* and *adjusted* configuration were measured using colorimeter. The measured values were in CIE XYZ, graphs comparing the values of X,Y and Z were plotted between the *plain* and *adjusted* backgrounds configuration. The figure 34 shows the relation between the X,Y and Z. The relation in the graph was found to be liner in each X,Y and Z for both OSTD. The linear equations obtained for the best fit line on the measured values are given in equations 7.1 for Lumisty OSTD and equations 7.2 for T-OLED OSTD. In each case the R^2 value were found to be almost 1.

$$\begin{aligned} AdjustedX &= 0.7041PlainX - 0.0012, R^2 = 0.9947 \\ AdjustedY &= 0.7066PlainY - 3E - 05, R^2 = 0.998 \\ AdjustedZ &= 0.626PlainZ + 0.0006, R^2 = 0.9961 \end{aligned} \tag{7.1}$$

$$\begin{aligned} AdjustedX &= 0.3478PlainX + 0.0008, R^2 = 0.9841 \\ AdjustedY &= 0.354PlainY + 0.002, R^2 = 0.9935 \\ AdjustedZ &= 0.2601PlainZ + 0.0101, R^2 = 0.9799 \end{aligned} \tag{7.2}$$

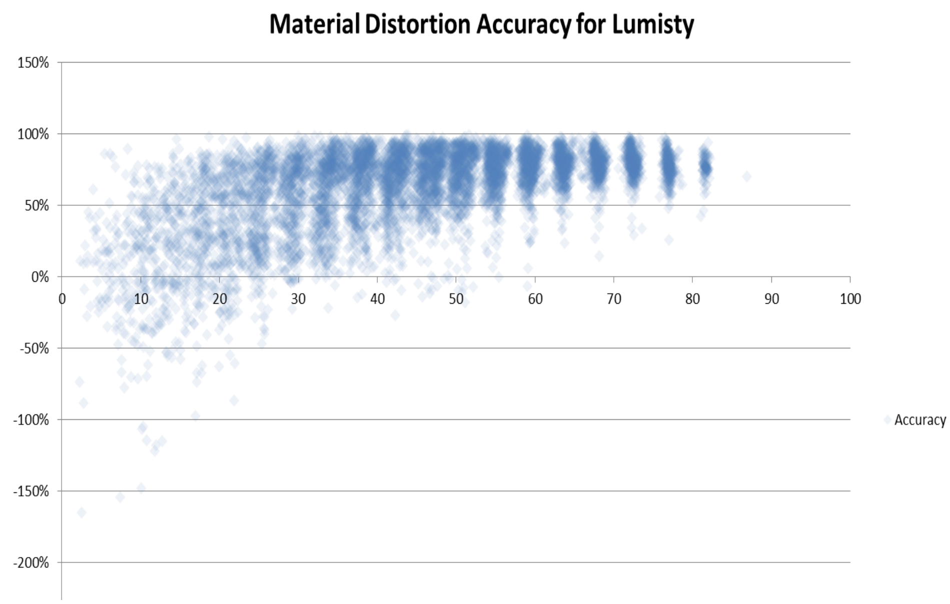
7.3 EXPLORATION

The linear equations given in 7.1 and 7.2 were used to *predict* the background values in *adjusted* configuration using the value in *plain* configuration. The predicted values were then converted in CIE LAB color space. The predicted *adjusted* colors (*pAdjusted*) accuracy percentage was calculated. Equations in 7.3 shows the step involved in the calculation of the accuracy percentage.

$$\begin{aligned}
PredictionDifference &= EqulideanDiff(MeauredadjustedColor, predictColor) \\
ActualDifference &= EqulideanDiff(MeauredadjustedColor, MeauredplainColor) \\
AccuracyPercentage &= \left(\frac{ActualDifference - PredictionDifference}{ActualDifference}\right) \times 100
\end{aligned} \tag{7.3}$$

Figure 35 shows the percentage accuracy obtained on Lumist and T-OLED OSTD for *pAdjusted* colors. The Lumisty OSTD showed a increased in accuracy with the increase in the color intensity. However this trend was not present in case of TOLED. The progressive nature of accuracy and the color intensity in Lumisty OSTD can be attributed to two facts; *First*, the low accuracy for darker color could be related to its color measurement precision. Hong et al. [8] has shown that precise capture of dark colors (L<50) is hard due to the low luminosity value of the colors. This can explain the reason for the low accuracy in darker colors in figure 35. *Second*, high intensity colors were present in *adjusted* configuration. Hong et al. [8] also shows that capture of light colors (L>50) values are precise. The presence of high intensity colors can be the reason behind accurate mapping between *plain* and *adjusted* configuration in figure 35- top (see figure 34 Lumisty Y - Max Value for intensity(Y) was 0.7 in *adjusted* configuration, this indicates that were high intensity color present). On contrary for T-OLED, the display material reduced the intensity of the color passing through it by more than 50% (Figure 34 T-OLED Y - Max Value for intensity was 0.35 in *adjusted* configuration, this shows that there were no color with intensity value higher than 0.35). The thickness of the T-OLED's acrylic cover was 9mm think, it visibly changed the background color's intensity. This might be the reason behind the inconsistent accuracy percentage in figure 35- bottom.

The lower transparency of TOLED darkens background colors. This is better suited for color correction as we have seen in chapter 5, colors were corrected the best for low intensity backgrounds. The degree of display opacity poses a challenge for how much accessibility and clarity is needed for either the background or the display content, and remains an open question. Potentially the linear functions as explained in this section can be used to predict the material distortion for bright backgrounds while for dark backgrounds colors *plain* conditions can be used as such. From what is shown in this thesis, further study and experimentation is needed to understand how each display material absorbs the color. Ideally once a strong correlation is found between *plain* and *adjusted* configuration the display manufacturer should be able to give the material distortion function.



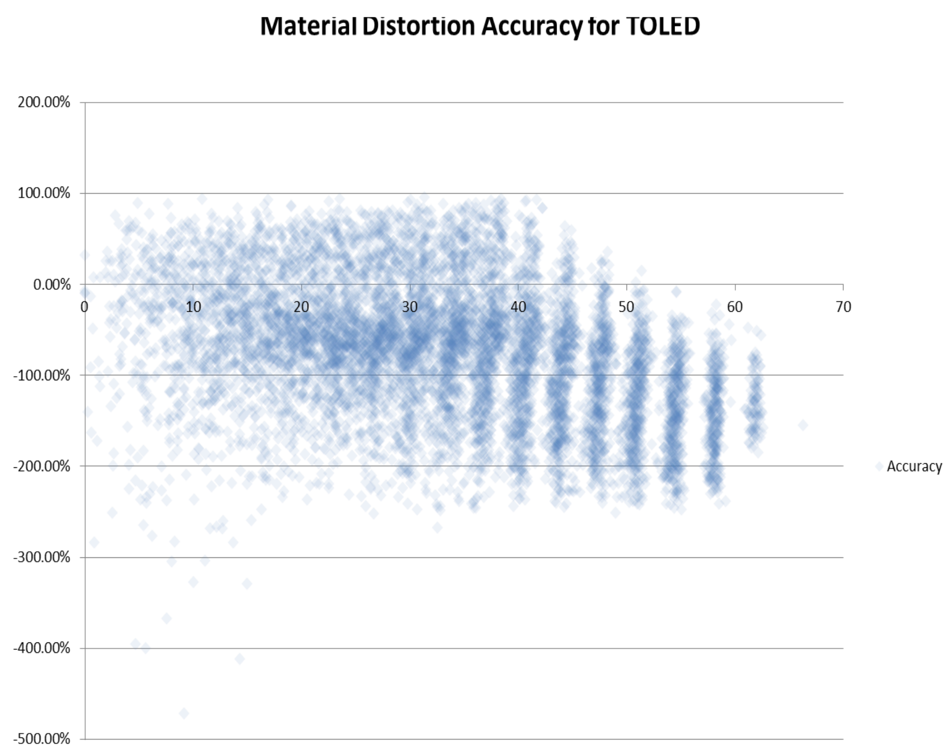


Figure 35: Accuracy percentage for the material distortion prediction - *pAdjusted* colors, with X-axis representing the *Adjusted* color's L value and Y axis the accuracy percentage

8

LIMITATIONS, RECOMMENDATIONS AND FUTURE WORK

This chapter looks into the limitations of BP based color correction. This chapter also gives recommendations on both digital content design and the OSTD hardware design taking the limitations and usability of OSTD into account. Finally potential future works are discussed.

8.1 LIMITATIONS OF BP COLOR CORRECTION

There are various limitation in using BP color correction. This section acknowledges and explains them with example. A major limitation of color correction is the fact that not all colors can be corrected in high intensity background condition, this holds true for BP color correction as well. Correction of a dark foreground color is impossible when it is shown an a high intensity background. Figure 36 is a good example of a dark color on a high intensity background, it can be seen when the background is as bright as white, the dark foreground color (red text) is not corrected and is invisible.

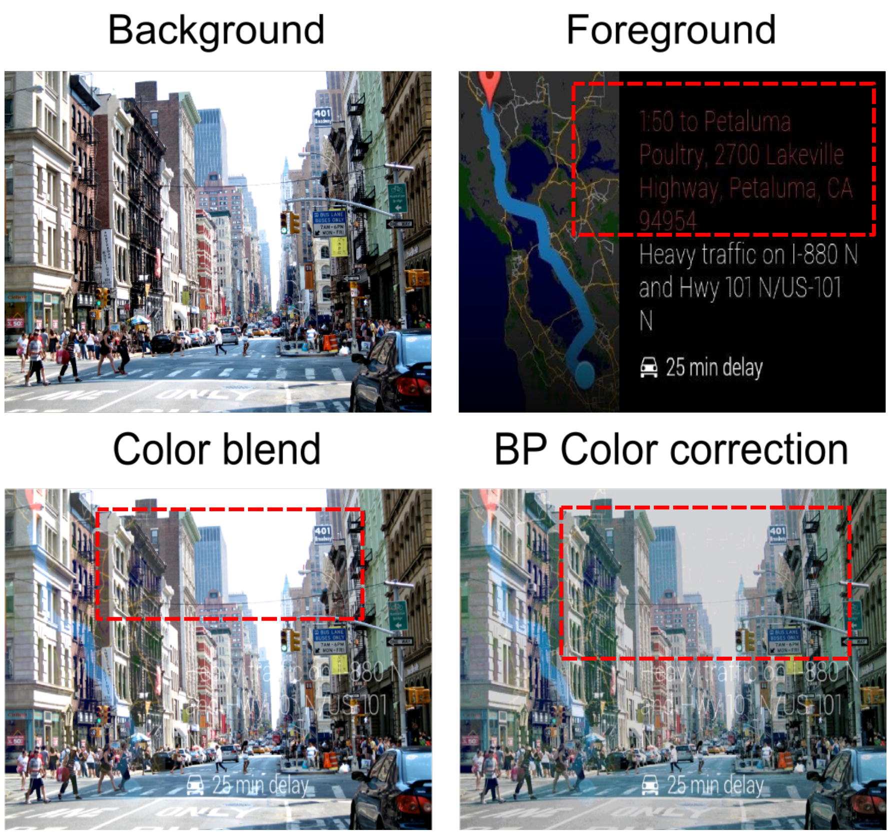


Figure 36: (Image best seen in color) Example showing inability of BP correction with a dark foreground text and high intensity background

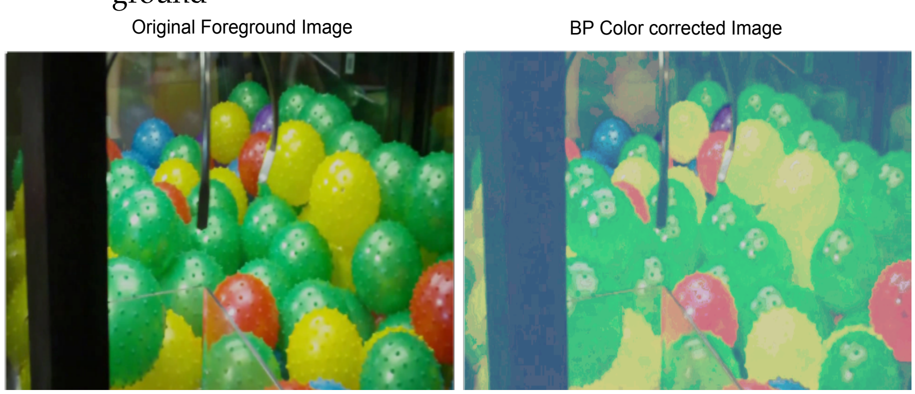


Figure 37: (Image best seen in color) Comparison of original image and the BP color corrected image for a blue background (notice the pixelation and loss of color resolution)

The second limitation is loss of color resolution. BP effectively reduces the continuous color space into a finite 8390 colors , with this the color resolution also gets reduces. It can be noted that the final blended image looks more pixelated than the intended foreground, this is also due to the fact that the BP has only 8390 colors. The pixelation and color resolution loss of the corrected image can be seen in figure 37. Even though the image preserves the colors on BP color correction, in the process the images loses color resolution.

Another major limitation is the execution time of BP color correction and inaccuracy of quick correction. The color correction at this execution time can be executed for a static background condition like a museum or shopping mall setting. Where the colors can be pre-computed before being corrected based on the background.However execution of BP color correction is not possible in real time for dynamic backgrounds.

8.2 design recommendation

Color correction depends largely on the background color's intensity, it would be wise to design digital contents based on the environment in which they will be deployed. Following a theme based design of digital contents for OSTD will be a wise way forward. Interface designers should study the intended normal usage conditions of their application (e.g. outdoors, forest or night time), in order to collect prevalent background colors in those conditions. Based on such a background color set, designers can analyse how correctable are alternative digital color palettes are and choose one palette which can be corrected the best for a given background condition.

It would also be wise to avoid dark colors to represent color critical contents like text, icons or labels in high intensity background conditions. Instead, usage of light color will be well suited for such cases. However in low intensity background conditions designers can be a bit more liberal in their color options. Figure 38 shows example of both background conditions. Dark digital color might be useful in maintaining contrast. Figure 39 shows how dark foreground(black) which was fully blended even upon BP color correction, still manages to creates contrast on the corrected image by picking the darker shades of the original background (figure 39 middle image). This effectively makes the corrected image look a bit darker than the color blend image. It is to be noted that these example figures are simulations, on a real OSTD the amount of contrast created will depend largely on the light source capacity to create these colors.

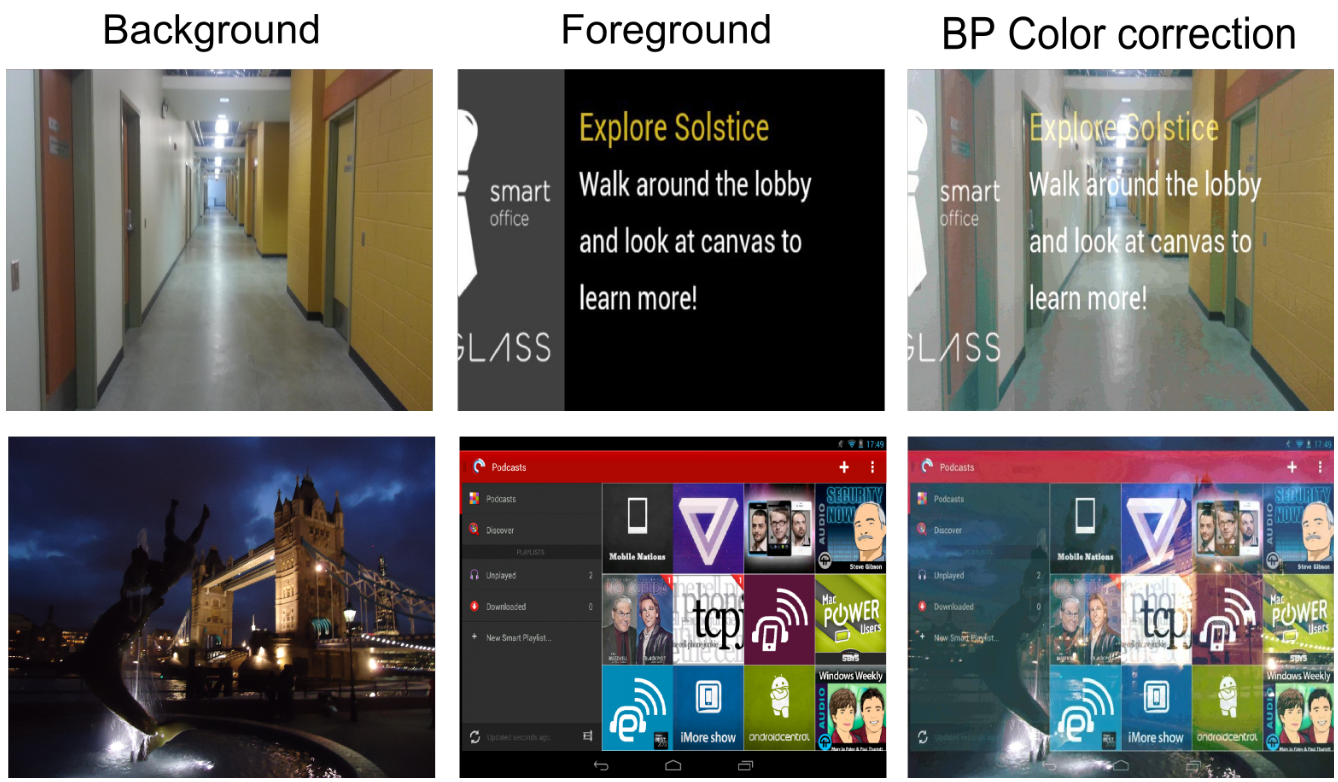


Figure 38: (Image best seen in color) Example of theme based UI design. Top images set is example for high intensity background where to design digital contents light colors are used and Bottom image set for low intensity background where to design digital contents various colors are used

Figure 39: (Image best seen in color) Example of usage of dark foreground color to create contrast. The image in the middle shows the BP correction result without the background. It can be seen darker shades of the background color are picked when the foreground is very dark (black).

As acknowledged in the precious section there is a heavy loss of color resolution and pixelation on use of BP based color correction. The loss of color resolution means BP color correction will not be able to correct high definition photos, images and videos with color resolution intact. However BP color correction can be very effective when used on digit contents like text, labels, maps and icons as they only need limited number of colors. designers can use BP to correct colors on such contents to maintain legibility.

8.3 display hardware recommendation

Work in this thesis shows how OSTD affects color blending both through its color profile (render distortion) and its impact on background colors (material distortion). On correction of colors using these two distortions, it was found that limited color profiles such as in the p2200 and T-OLED displays guaranties better BP-based color correction. However, with the limitation of pixelation and loss of color resolution such displays provide a limited color experience for digital contents. On contrary, full color displays like the p3700 can provide richer color resolution. This might come at the cost of lower correction accuracy.

It was also found color correction on a display (TOLED) which has lower transparency produced better results. A promising OSTD hardware configuration would be a full color displays with a lower level of transparency. Lower transparency is proportional to the effect of material distortion. Lower transparency darkens background colors and might be better suited for color correction. The degree of display opacity/ transparency ideally should dependent on a informed trade off between the accessibility and clarity needed for either the background or the display content. The ideal match of the profile size and the material's transparency remains an open question.

Alternatively, an OSTD can provide a way to control the level of transparency at a pixel level with the use of a spatial light modulator as shown by Kiyokawa et al. [11]. On top of this hardware enhancement rendering pipeline could rely on BP-based color correction for normal darker backgrounds and SLM's (spatial light modulator) can become active in blocking the background light, as the background's intensity goes beyond the algorithm's correction limit.

8.4 FUTURE WORK

The work in this thesis lays down the foundation in terms of understanding color blending and its correction in additive OSTD. There is a very large potential to build on top of this work with many commercial OSTD devices hitting the consumer markets. This section will give out some ideas and necessities which are needed in taking this research forward.

One of the major target for future work is development of a real time color correction algorithm top of this research work. Quick correction is a step in that direction however lacks accuracy. Once a real time algorithm is developed then it can be build into the rendering pipeline in off the shelf head mounted OSTD displays. In the wild implementation of this system with user participants and for different display hardware (i.e. head-mounted displays and window size transparent displays) will result in robust OSTD.

Future research could also look into the use of SLM's and a practical way for implementing a OSTD hardware which works as a hybrid of the SLM and color correction.

Further work is also needed in characterizing the material distortion along the lines expressed in chapter 7.

9

CONCLUSION

The work in this thesis aims to be the basic ground work on understanding color blending and factors involved in color correction on a additive OSTD.

This thesis presents a color correction approach for additive optical see-through displays (OSTD) based on two color distortions introduced by the display: the render and material distortions. The work proposes a color blending model for additive OSTD which accounts for both the distortions. The Binned-Profile (BP) method is proposed to address the render distortion. The BP method which uses the BP, describes the way a particular display renders a standard set of 8390 colors. For the second distortion colorimetric measurements of how background colors are seen through the display material were used.

The BP-method was validated by measuring the accuracy of its color blend predictions on three different OSTD against other known methods. The results show that the BP method outperforms other methods, and indicates that render distortion is the dominating factor in color blend predictions.

A color correction algorithm based on the BP-method called BP-color correction is devised and its correction capacity analysed using a wide range of background and display colors. The analysis was also done on all three OSTD. BP color correction was compared to existing work and was shown to produce better result. On color analysis the results showed the BP-based color correction works best for displays with low color resolution. For displays with high color resolution the results show that colors can be better corrected for low intensity backgrounds, and that for high intensity backgrounds light neutrals and light CyanBlue colors can be corrected best.The results are reported both graphically (through vertical histograms and heat-maps) and quantitatively.

An quicker variation of color correction was developed based on BP, called quick correction. Quick corrections finds an approximate solution at a faster rate than BP color correction. Quick corrections accuracy is compared with BP color correction. Results showed that quick correction did find a color similar hue to that of BP color correction. Quick correction execution time is 99% faster than BP color correction. This thesis also has provided basic findings for material distortion, explaining how it was recoded for the different display material used.

The work also discusses the limitation of BP- based color correction. This work also recommends design ideas for digital contents on OSTD and configurations for the display hardware.

APPENDIX A

A.1 CAT MATRICES

This section gives the final CAT transformed matrices for different displays used.

A.1.1 *P2200:*

$$CATBradford = \begin{pmatrix} 3.81754059 & -0.149043619 & -0.044288985 \\ 0.178356377 & 3.418828937 & -0.025150971 \\ -0.727790647 & -1.788890974 & 3.714721549 \\ \end{pmatrix}$$

$$CATVonKries = \begin{pmatrix} 3.666332229 & 0.400609337 & -0.283244448 \\ 0.044004094 & 3.517532015 & -0.008877716 \\ 0 & 0 & 2.26352454 \\ \end{pmatrix}$$

$$CATScaling = \begin{pmatrix} 3.578954107 & 0 & 0 \\ 0 & 3.54381215 & 0 \\ 0 & 0 & 2.26352454 \\ \end{pmatrix}$$

A.1.2 *TOLED:*

$$CATBradford = \begin{pmatrix} 2.53964670 & -0.06984405 & 0.01271228 \\ -0.03148576 & 2.55542751 & 0.00721909 \\ 0.59612492 & -0.16841149 & 2.50520155 \\ \end{pmatrix}$$

$$CATVonKries = \begin{pmatrix} 2.50176106 & -0.1046120 & 0.08907704 \\ -0.01149088 & 2.54061759 & 0.00231782 \\ 0 & 0 & 2.9429269 \\ \end{pmatrix}$$

$$CATScaling = \begin{pmatrix} 2.47993550 & 0 & 0 \\ 0 & 2.531639 & 0 \\ 0 & 0 & 2.94292695 \\ \end{pmatrix}$$

A.1.3 *p3700:*

$$CATBradford, CATVonKries, CATScaling = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \end{pmatrix}$$

BIBLIOGRAPHY

[1] M. Ashdown, T. Okabe, I. Sato, and Y. Sato. Robust content-dependent photometric projector compensation. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, pages 6–6, 2006. (Cited on pages 13, 14, 45, and 46.)

[2] O. Bimber, A. Emmerling, and T. Klemmer. Embedded entertainment with smart projectors. *Computer*, 38(1):48–55, 2005. (Cited on pages 4, 12, 14, 45, 46, and 75.)

[3] O. Cakmakci, Yonggang Ha, and J.P. Rolland. A compact optical see-through head-worn display with occlusion support. In *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*, pages 16–25, 2004. (Cited on pages xii, 2, and 4.)

[4] J.L. Gabbard, J.E. Swan, J. Zedlitz, and Woodrow W. Winchester. More than meets the eye: An engineering study to empirically examine the blending of real and virtual color spaces. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 79–86, 2010. (Cited on pages 1, 3, 5, 19, and 21.)

[5] M.D. Grossberg, H. Peri, S.K. Nayar, and P.N. Belhumeur. Making one object look like another: controlling appearance using a projector-camera system. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–452–I–459 Vol.1, 2004. (Cited on page 13.)

[6] G.E. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(3):267–276, 1994. (Cited on page 75.)

[7] Jeffrey Heer and Maureen Stone. Color naming models for color selection, image editing and palette design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1007–1016, New York, NY, USA, 2012. ACM. (Cited on page 24.)

[8] Guowei Hong, M. Ronnier Luo, and Peter A. Rhodes. A study of digital camera colorimetric characterization based on polyno-

mial modeling. volume 26, pages 76–84. John Wiley and Sons, Inc., 2001. (Cited on pages 75 and 79.)

[9] Steven J. Kerr, Mark D. Rice, Yinquan Teo, Marcus Wan, Yian Ling Cheong, Jamie Ng, Lillian Ng-Thamrin, Thant Thura-Myo, and Dominic Wren. Wearable mobile augmented reality: evaluating outdoor user experience. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, VRCAI '11, pages 209–216, New York, NY, USA, 2011. ACM. (Cited on pages 3 and 4.)

[10] Kiyoshi Kiyokawa, Mark Billinghurst, Bruce Campbell, and Eric Woods. An occlusion-capable optical see-through head mount display for supporting co-located collaboration. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '03, pages 133–, Washington, DC, USA, 2003. IEEE Computer Society. (Cited on pages 2 and 4.)

[11] Kiyoshi Kiyokawa, Yoshinori Kurata, and Hiroyuki Ohno. An optical see-through display for mutual occlusion with a real-time stereovision system. *Computers and Graphics*, 25(5):765 – 779, 2001. <ce:title>Mixed realities - beyond conventions</ce:title>. (Cited on pages 4 and 88.)

[12] Kiyoshi Kiyokawa, Hiroyuki Ohno, and Yoshinori Kurata. Occlusive optical see-through displays in a collaborative setup. In *ACM SIGGRAPH 2002 conference abstracts and applications*, SIGGRAPH '02, pages 74–74, New York, NY, USA, 2002. ACM. (Cited on pages 2 and 4.)

[13] E. Kruijff, J.E. Swan, and S. Feiner. Perceptual issues in augmented reality revisited. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 3–12, 2010. (Cited on page 3.)

[14] Lenovo. Lenovo s800:full phone specifications. `http://www.gsmarena.com/lenovo_s800-4862.php.`, 2013. Accessed: 2013-04-23. (Cited on pages vii and 18.)

[15] S. Lin, Jinwei Gu, S. Yamazaki, and Heung-Yeung Shum. Radiometric calibration from a single image. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–938–II–945 Vol.2, 2004. (Cited on page 75.)

[16] Lindbloom. Chromatic adaptive transforamtion. `http://www.brucelindbloom.com/index.html?Eqn_ChromAdapt.html`, 2013. Accessed: 2013-04-23. (Cited on pages xii and 27.)

[17] Lumisty. Lumisty lumisty mfz-2555 view angles. http://www.glassfilmenterprises.com/lumisty.htm, 2013. Accessed: 2013-04-23. (Cited on pages 16 and 17.)

[18] M Mahy, L Van Eycken, and A Oosterlinck. Evaluation of uniform colour spaces developed after the adoption of cielab and cieluv. *Color Research and Application*, 19(2):105–121, 1994. (Cited on pages xii and 24.)

[19] H. McCamy, C. S.and Marcus and J. G. Davidson. A color-rendition chart. *Journal of Applied Photographic Engineering*, 2(3):95–99, 1976. (Cited on pages 19 and 29.)

[20] C. Menk and R. Koch. Interactive visualization technique for truthful color reproduction in spatial augmented reality applications. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 157–164, 2011. (Cited on pages 13, 14, 45, and 46.)

[21] T. Mitsunaga and S.K. Nayar. Radiometric self calibration. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages –380 Vol. 1, 1999. (Cited on page 75.)

[22] Shree K. Nayar, Harish Peri, Michael D. Grossberg, and Peter N. Belhumeur. A projection system with radiometric compensation for screen imperfections. 2003. (Cited on pages 4, 12, 45, and 46.)

[23] T. J. Pingel and K. C. Clarke. Assessing the usability of a wearable computer system for outdoor pedestrian navigation. *Autocarto*, 2005. (Cited on pages xii and 4.)

[24] A.B. Sekuler and S.E. Palmer. Perception of partly occluded objects: A microgenetic analysis. *Journal of Experimental Psychology*, 121(1):95–111, 1992. (Cited on page 4.)

[25] Sabine E. Susstrunk, Jack M. Holm, and Graham D. Finlayson. Chromatic adaptation performance of different rgb sensors. pages 172–183, 2000. (Cited on pages xii, 10, and 27.)

[26] Kohei Tanaka, Yasue Kishino, Masakazu Miyamae, Tsutomu Terada, and Shojiro Nishio. An information layout method for an optical see-through head mounted display focusing on the viewability. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pages 139–142, Washington, DC, USA, 2008. IEEE Computer Society. (Cited on pages 2 and 4.)

[27] Christian Weiland, Anne-Kathrin Braun, and Wolfgang Heiden. Colorimetric and photometric compensation for optical see-through displays. In *Universal Access in Human-Computer Interaction.*, volume 5615, pages 603–612, 2009. (Cited on pages xii, 10, 14, 26, 38, 45, 51, 52, 53, 57, and 75.)

[28] Ya Zhou, Jin-Tao Ma, Qun Hao, Hong Wang, and Xian-Peng Liu. A novel optical see-through head-mounted display with occlusion and intensity matching support. In *Technologies for E-Learning and Digital Entertainment*, volume 4469 of *Lecture Notes in Computer Science*, pages 56–62. Springer Berlin Heidelberg, 2007. (Cited on pages 2 and 4.)

COLOPHON

This thesis was typeset with the `pdflatex` $\LaTeX 2_\varepsilon$ interpreter using Hermann Zapf's *Palatino* type face for text and math and *Euler* for chapter numbers. The listings were set in *Bera Mono*.

The typographic style of the thesis was based on André Miede's wonderful `classicthesis` LaTeX style available from CTAN. My modifications were limited to those required to satisfy the constraints imposed by my university, mainly 12pt font on letter-size paper with extra leading. Miede's original style was inspired by Robert Bringhurst's classic *The Elements of Typographic Style* [**?**]. I hope my naïve, yet carefully considered changes are consistent with Miede's original intentions.

Final Version as of January 3, 2014 at 17:13.