ANALYZING COLOR IMAGING FAILURE ON CONSUMER CAMERAS

SAIKIRAN KUMAR TEDLA

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

JULY 2022

Abstract

There are currently many efforts to use consumer-grade cameras for home-based health and wellness monitoring. Such applications rely on users to use their personal cameras to capture images for analysis in a home environment. When color is a primary feature for diagnostic algorithms, the camera requires color calibration to ensure accurate color measurements. Given the importance of such diagnostic tests for the users' health and well-being, it is important to understand the conditions in which color calibration may fail. To this end, we analyzed a wide range of camera sensors and environmental lighting to determine (1): how often color calibration failure is likely to occur; and (2) the underlying reasons for failure. Our analysis shows that in well-lit environments, it is rare to encounter a camera sensor and lighting condition combination that results in color imaging failure. Moreover, when color imaging does fail, the cause is almost always attributed to spectral poor environmental lighting and not the camera sensor. We believe this finding is useful for scientists and engineers developing color-based applications with consumer-grade cameras.

Acknowledgements

This thesis is dedicated to my family and friends; I aspire to be like them.

Thank you to my supervisor, Dr. Michael Brown, for understanding my passions and providing me with a project that allows me to explore computer vision in a practical manner. Dr. Brown is a role model for me and I am excited for our future work.

I would like to thank Dr. Konstantinos Derpanis and Dr. Richard Murray for being a part of my committee. I would like to thank Dr. Terrance Boult who gave me my passion for research when I was unsure about my interests. Thank you to Vivian Wang and Maitri Patel for supporting my research.

Thank you to my family who have been gracious at all times: Shravan Tedla, Suneetha Tedla, Siddhu Tedla, Sukumar Tedla, Rajyam Tedla, Sashivarna Gunda, Lingiah Gunda, Srinu Gunda and family, Kumar Gunda and family, Venkana Gunda and family, Laxmi Cheela and family, Sridhar Tedla and family, and the Colorado Springs Telugu family.

Thank you to my mentors: Ramesh Kanekal, Dr. Chad Mello, Brian Olson, Vivian Wang, Aaron Rowe, Amit Tuteja, Mayank Bansal, Rajani Maddhi, Justin McDonald, and Sandeep Mekala.

Thank you to all my friends who have been nothing but extraordinary: Sai Maddhi, Jason Alexander, Yashi Uppalapati, Rahul Shenoy, Sumanth Talanki, Kobey King, Rice Province, Rajat Ajayakumar, Jennifer Jaison, Sina Adineh, Mohammed Aboutaleb, Elias Lorigados, Yoland El-Hajj, Andy Li, and Karmon Dhillon.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Problem

While personal cameras are primarily used for photographic purposes, a growing number of applications aim to use personal cameras for tasks related to health and wellness. In such cases, the camera is treated as scientific instrument. Examples include applications for monitoring blood glucose and pH levels [1, 2], measuring blood pressure [3], and skin analysis [4, 5]. This trend is becoming increasingly popular as consumers now have easy access to a camera integrated into their smartphones. When color is required by such applications for image analysis, it is important to perform a color calibration procedure to ensure accurate color measurements [6]. It is possible that the color calibration may fail as shown in Fig. 1.1. When failure does occur, it is important for the mobile app to react and direct the user towards corrective action.

To motivate our work, it is necessary to explain why color calibration is required. Con-

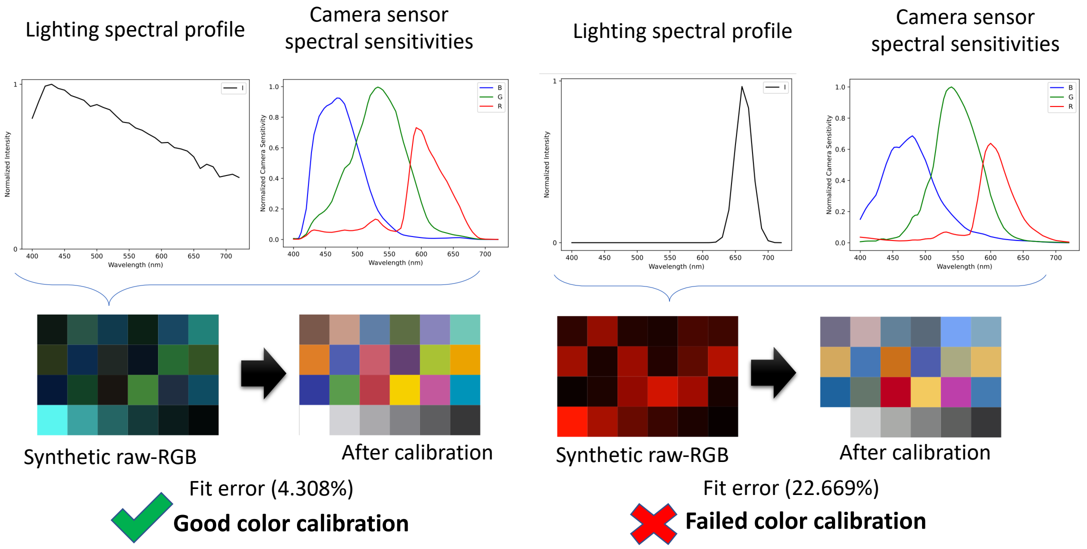


Figure 1.1: Shown are an example of a successful color-calibrated image and an image with failed color calibration. The images are synthetically generated X-Rite color charts captured by a camera under two different lighting conditions: D65 lighting and red LED lighting. Generation procedure of these images will be described in Sec. 3. The fit errors describe the percent error of using a linear correction matrix on the raw-RGB image. This paper is interested in determining how often such failures occur and the underlying reason for failure—namely, the camera sensor spectral sensitivities, spectral properties of the light source, or degenerate combinations of sensor and lighting).

sumer cameras apply a series of image processing steps to enhance the aesthetic quality of the captured images. Such in-camera processing can be scene-specific and vary based on camera settings [7, 8].

Fig.1.2-(A) shows a diagram where two cameras capturing a color chart under the same illumination. In this example, the two cameras have sensors with different spectral sensitivities. The image captured directly by the sensor is referred to as a raw-RGB image. Due to the different spectral sensitivities, the raw-RGB images will be different.

Consumer cameras further process the raw-RGB images using specialized hardware called an image signal processor (ISP). The ISP applies image and color manipulation to convert the raw-RGB image into a standard color space, namely standard RGB (sRGB). However, the ISP also applies photo-finishing to enhance the visual appearance of the image. As a result, the images outputted by the cameras will be visually different as shown in Fig. 1.2-(B).

The color variations from different cameras is a problem for health and medical applications. As a result, it is necessary to bypass the onboard camera image manipulation by directly capturing the sensor's raw-RGB image [6]. However, the raw-RGB colors are in a sensor-specific color space due to the sensors' different spectral sensitivities. As a result, a colorimetric mapping needs to be computed at capture time via a color calibration procedure. This calibration not only compensates for the sensor's spectral sensitivities but also implicitly includes white-balancing to accommodate for different environmental lighting. Color calibration is typically performed by placing a calibration chart with color patches with known colorimetric values (e.g., in CIE XYZ) in the scene. By imaging the color calibration chart, a colorimetric mapping for the entire image can be estimated based on correspondences between the color patches and their observed raw-RGB values using standard numerical methods. A typical camera calibration procedure is shown in Fig. 1.2-(C).

Colorimetric calibration is a well-studied problem with many proposed solutions regarding the type of mapping used to convert from sensor-specific raw-RGB values to a device-independent perceptual color space, such as CIE XYZ (e.g., [9, 10, 6]). Using a color chart also makes it easy to detect when color imaging fails. This can be done by simple analysis of the residual error between the transformed raw-RGB values using the estimated mapping and target CIE XYZ values of the chart.

The work in this thesis is *not* focused on how to detect color imaging failure; instead, the focus is on understanding how likely such failures are and the underlying cause when failures occur. As previously mentioned, such analysis is important to direct users toward corrective actions in the event that color calibration fails. For example, should the home-care app direct a user to seek a different camera or another room with different lighting, or both?

1.2 Thesis Contribution

Assuming a well-lit environment and proper exposure and focus settings, we perform analysis to determine what factors lead to color imaging failure. We are particularly interested to determine if specific illumination-sensor pairs lead to unexpected color imaging failure – namely, are there specific consumer sensors and environment lighting combinations that result in poor color calibration.

To this end, we perform experiments that simulate color imaging using 97 consumer camera sensors (DSLR and smartphone) whose spectral sensitivities have been measured. We simulate these sensors' ability to perform colorimetric calibration with 108 commercial and environmental light sources with different spectral profiles. Our finding reveals that consumer-grade sensors and typical lighting environments are suitable for color imaging. When color imaging does fail, it is inevitably caused by unusual lights with narrow-band spectral profiles (e.g., single color LEDs). For the sake of completeness, we em-

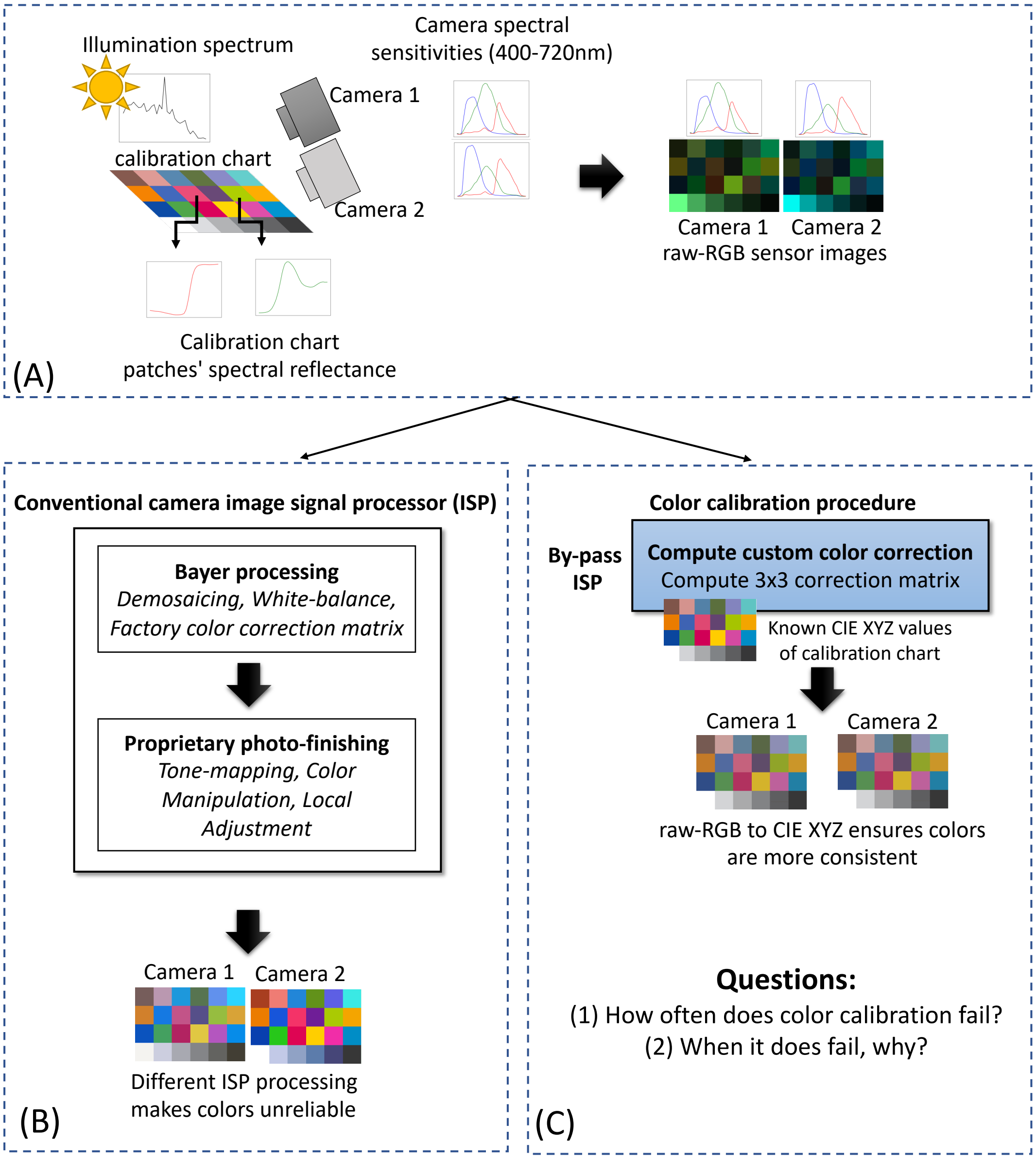

Illumination spectrum
Camera spectral sensitivities (400-720nm)
Camera 1
Camera 2
calibration chart
Camera 1 Camera 2
raw-RGB sensor images
Calibration chart patches' spectral reflectance
(A)
Conventional camera image signal processor (ISP)
Bayer processing
Demosaicing, White-balance, Factory color correction matrix
Proprietary photo-finishing
Tone-mapping, Color Manipulation, Local Adjustment
Camera 1
Camera 2
Different ISP processing makes colors unreliable
(B)
Color calibration procedure
By-pass ISP
Compute custom color correction
Compute 3x3 correction matrix
Known CIE XYZ values of calibration chart
Camera 1
Camera 2
raw-RGB to CIE XYZ ensures colors are more consistent
Questions:
(1) How often does color calibration fail?
(2) When it does fail, why?
(C)

Figure 1.2: This figure illustrates the need for colorimetric calibration. (A) Shows the image formation for two different cameras. Since the spectral sensitivity of the sensors are different, the resulting raw-RGB images are will be different. (B) While the conventional image processing pipeline does apply color correction to account for the device-dependent color space, it also applies proprietary photo-finishing. As a result, the images of the same scene are still different. (C) Shows a colorimetric calibration procedure required to ensure scenes from different camera sensors appear the same. We are interested in determining when the process in (C) fails.

pirically examine the same phenomenon using a small sample of smartphone cameras and commercial lighting using a custom calibration chart. Again, these real-world experiments indicate that color calibration is typically successful and failure is attributed to unusual spectral-poor lighting. This work in this thesis has been published in the Journal of the Optical Society of America A (JOSA-A) [11].

Chapter 2

Background

In this chapter, we provide a background on color and in-camera processing as it pertains to our work. Related work, in regards to color calibration, is also discussed.

2.1 Color

To first understand how color failure occurs, it is necessary to establish a basic understanding of color and how cameras interpret the color of a scene.

The physical world does not have color. Color is the human interpretation of the electromagnetic radiation that exists in the physical world. Human perception is sensitive to wavelengths between 380nm and 720nm [12]. Fig. 2.1 shows how the range of human perception is only a small range of the electromagnetic spectrum. Physical objects reflect (or emit) light over the visible spectrum. This can be represented as a spectral power distribution (SPD) as shown in Fig. 2.1. In this example, the SPD of three items (candy, wool, flower) are shown. It is important to note that Fig. 2.1 is assuming uniform illumination across the spectrum.

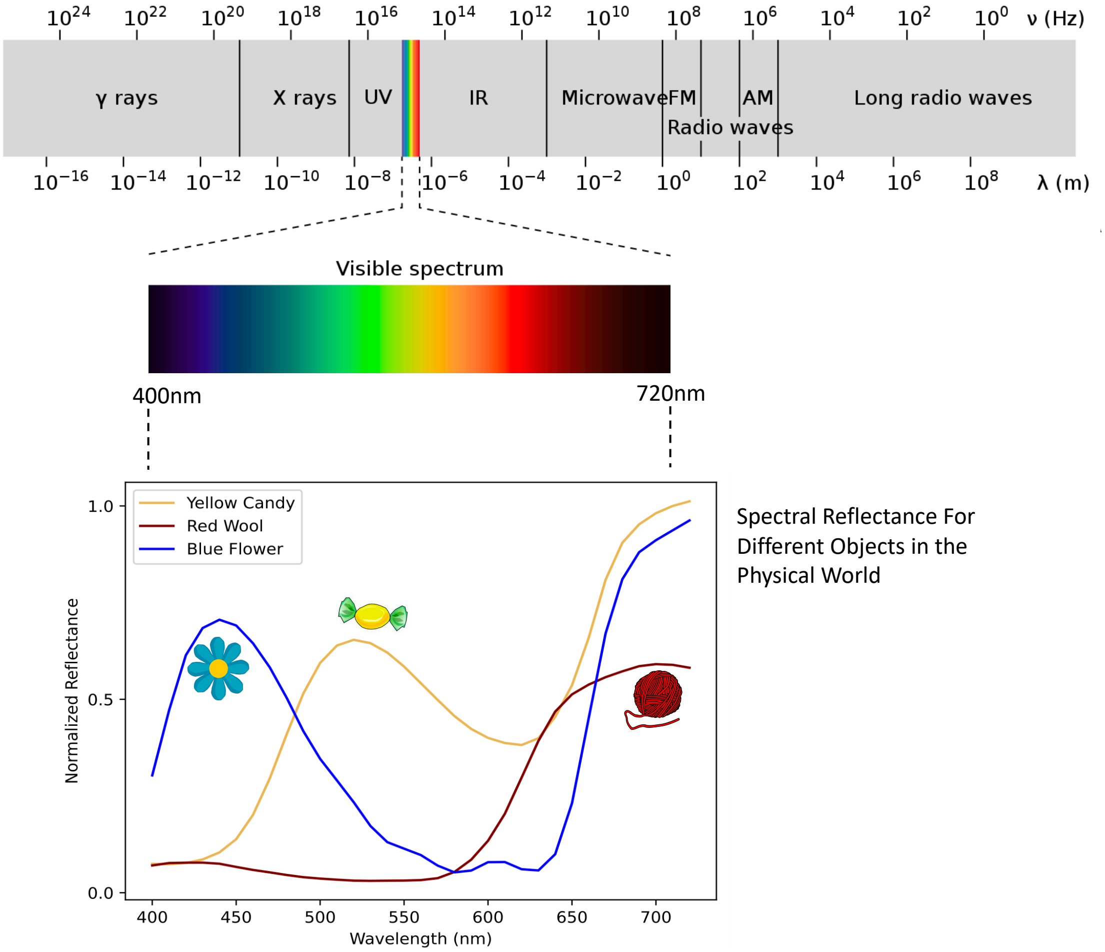


Figure 2.1: This figure illustrates how color is electromagnetic radiation. We show the visible spectra of three different objects: yellow candy, red wool, and a blue flower. Top part of figure adapted from [13] under CC3.0. Pictures of objects are public domain or CC1.0 [14–16].

Color is defined within the context of human perception [17]. The interpretation of color starts from the biological response of the cone cells in the eyes to incoming scene radiance. There are three types of cone cells in the eye: long, medium, and short (often referred to as the LMS cones). The name of the cone represents the types of wavelengths it is more sensitive to [18, 19] – e.g., a short cone is sensitive to the shorter wavelengths (that we associate with blue). Each of the cone types integrates the wavelengths it is sensitive too. The interpretation of color is based of the ratio of long, medium, and short responses [19]. Fig. 2.2 of the sensitivity shows both the location of cones in the eye and also the sensitivity functions of the cones. The LMS cone response to the SPD of the candy, wool, and flower, produce the sensations that allow us to perceive the candy as yellow, the wool as red, and the flower as blue.

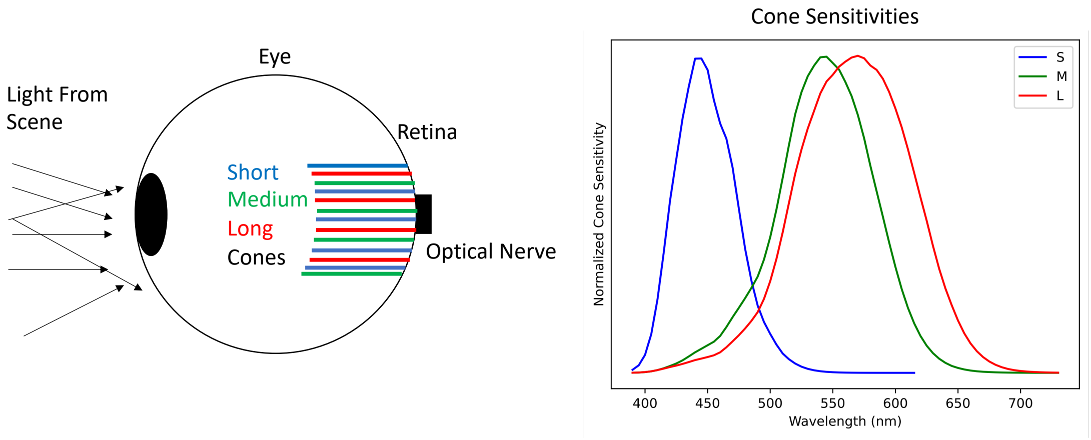


Figure 2.2: This figure illustrates the location of cones in the eyes and the sensitivity functions of those cones.

The Commission Internationale de L'éclairage (CIE)[1] is the long-standing body tasked with establishing standards for color and illumination. Before the biology of the eye was understood (i.e., the LMS responses), experiments were performed to develop a set of standard sensitivity functions for human color perception known as the CIE 1931 XYZ sensitivity functions.

[1] English translation: International commission on illumination

Fig. 2.3 shows the sensitivity functions established by CIE XYZ. The XYZ matching function have several desirable properties for colorimetric analysis. Two important properties of the XYZ space include non-negative values in the matching functions and that the Y matching function corresponds to luminance (perceived brightness).

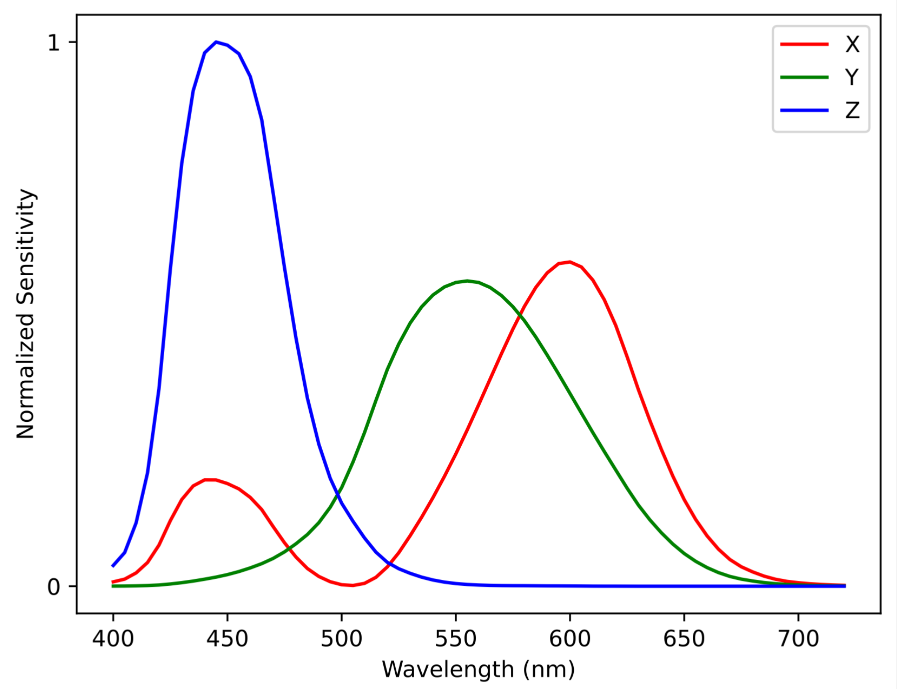


Figure 2.3: This figure shows he sensitivity curves for the CIE XYZ space.

We can think of the CIE XYZ curves as representing the spectral sensitivities of a "human" camera sensor. Essentially, CIE XYZ provides a device-independent color space that can be used as a standard to tie together different devices, including camera sensors [17].

To mimic the tri-stimulus nature of human vision, camera sensors use three color filters that are sensitive to different wavelengths. These are often referred to as red, green, and blue (RGB). Different sensor make and models use different color sensitivities as shown in Fig. 2.4. These means that the raw-in this exaues captured by a sensor are in a sensor-specific color space defined by the filters used.

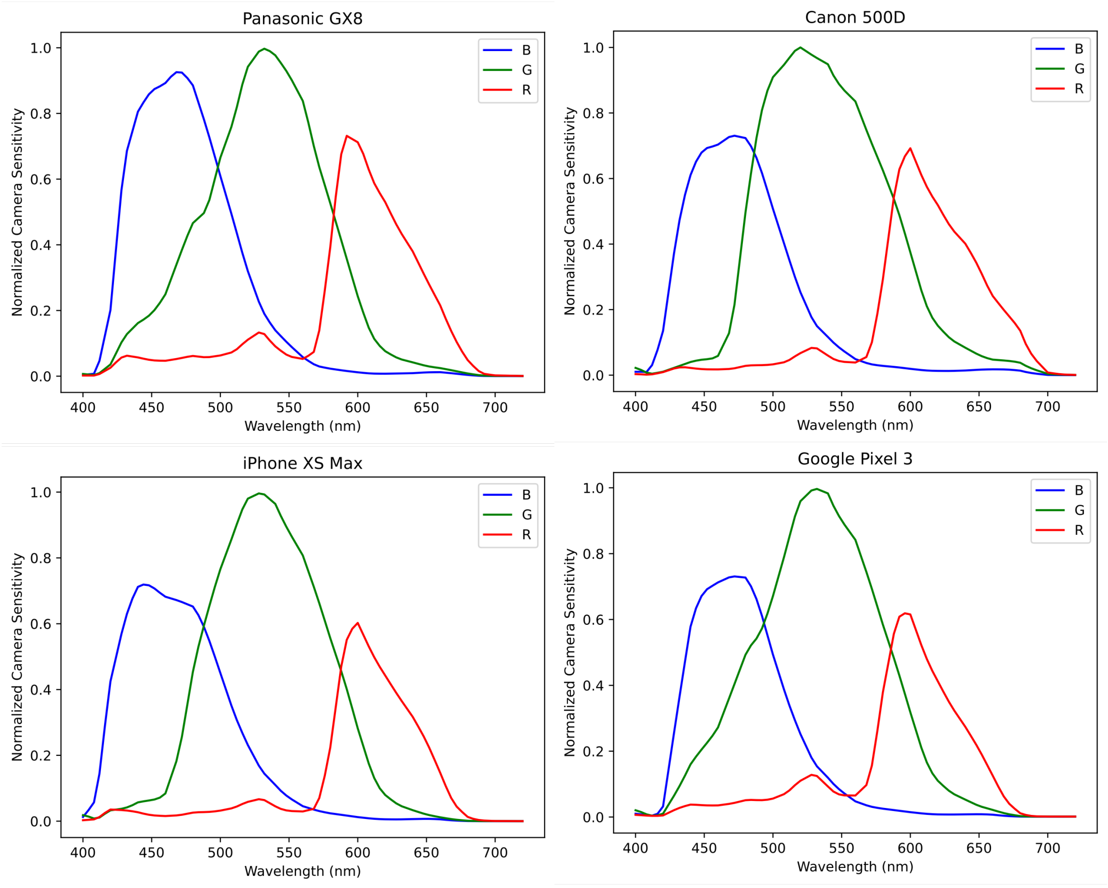


Figure 2.4: This figure shows the sensitivity functions from different cameras including mobile phones and DSLR cameras.

Colorimetric mapping is the process of mapping a sensor-specific RGB responses to the device-indepenent CIE XYZ color space [20]. Fig. 2.5 shows a camera response to the SPD of the patches on standard X-Rite color rendition chart and the colorimetric mapping to the common CIE XYZ space.

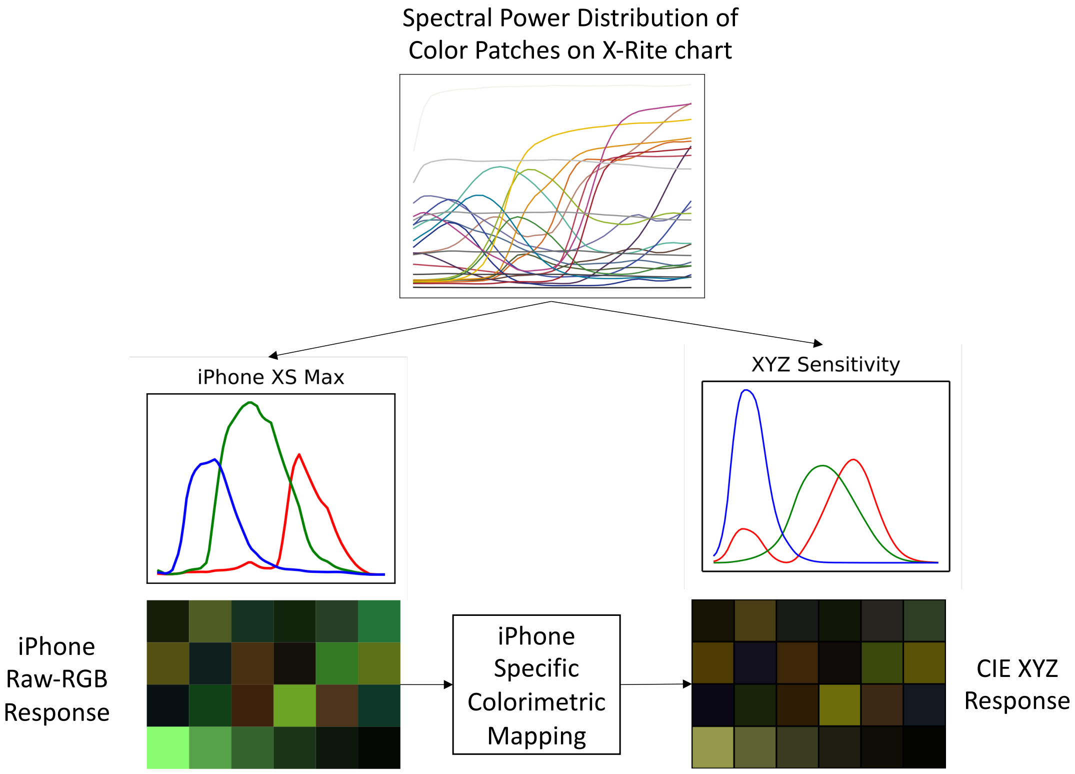


Figure 2.5: This figure shows how the colorimetric mapping step must be computed from a camera-specific RGB pace into the common XYZ space. The top of the figure shows the power spectral density of the X-Rite standard color chart under a uniform illumination. The camera specific RGB response to this input is shown using the iPhone XS Max sensitivity curves. The XYZ response is also shown using the standard CIE XYZ curves.

Computing a colorimetric mapping from a camera specific raw-RGB space to to colorimetric linear space like XYZ is a common problem that has been explored in [20, 21]. There are many methods to try to and find mappings between illumination and camera specific raw-RGB spaces to CIE XYZ given a known color chart in the scene. This problem is primarily addressed using techniques including linear [22]/polynomial [10] fits and look up table [23].

As mentioned in Chapter 1, the focus of this thesis is not computing a colorimetric mapping, but understanding when such mapping fail. To this end, we need to examine another factor related to color – scene illumination.

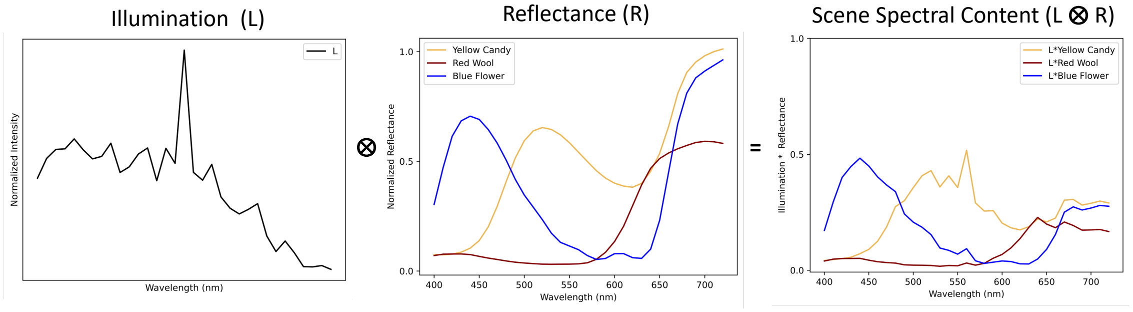


Figure 2.6: This figure shows how the scene spectral content is a combination of illumination and reflectance.

The example of an object's SPD shown in Fig. 2.1 was not entirely correct. The electromagnetic radiation that enters our camera or eye from a scene is a combination of both illumination and reflectance. The example in Fig. 2.1 assumed an ideal condition of pure white-light. In reality, we observe the physical world under a wide variety of scene illumination. We can factor in scene lighting by multiplying the lighting's SPD, denoted by $\mathbf{L}$, and the object's SPD recording reflectance, denoted as $\mathbf{R}$, as follows:

$$\text{Scene Spectral Content} = \mathbf{L} \cdot \mathbf{R}, \tag{2.1}$$

where $\cdot$ represents pointwise multiplication of each wavelength. Fig. 2.6 shows an example with two different illuminations L_1 and L_2.

One might expect an observer to perceive the colors differently under varying illuminations. However, the human vision system has the impressive ability to view objects as the same color regardless of the illumination; this ability is known as color constancy [24]. However, camera sensors do not perform color constancy. As a result, the scene illumination will need to be compensated for via a white-balance procedure. White balance is part of the camera pipeline which is described in the following section.

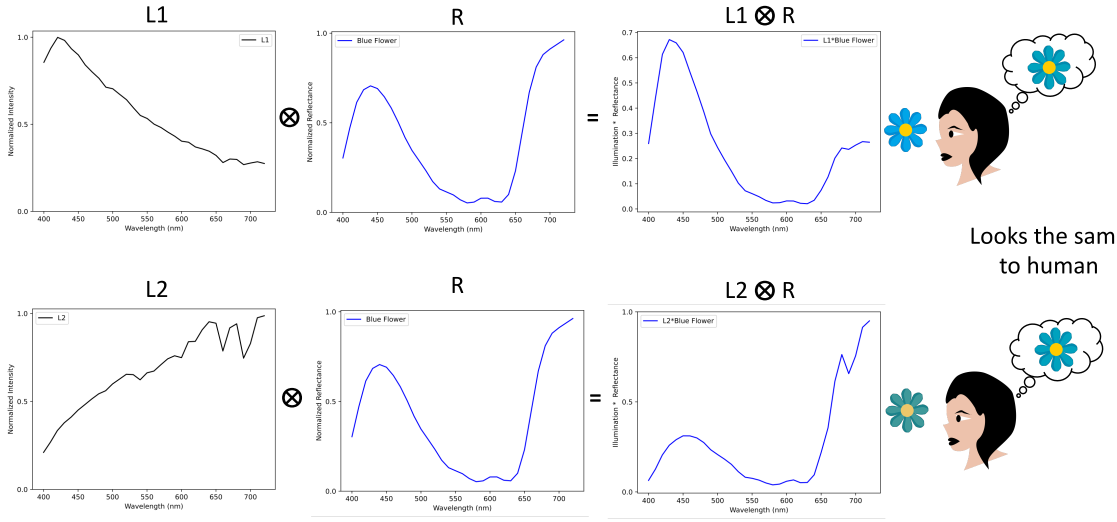


Figure 2.7: This figure (Adapted from [25]) shows the spectral content of a flower under two different illuminations. Although two different spectrums reach a human's eye, they are interpreted as the same color by a human observer due to our ability of color constancy. Image of face and flower are public domain images [26, 14].

2.2 Camera Imaging Pipeline

In the previous section, we established the basics of how humans interpret color, and standard representations, such as CIE XYZ. We also saw that for a camera, the the colorimetric mapping problem has two components: scene illumination and camera spectral sensitivity. This thesis tries to understand how illumination and camera sensitivity can effect the colorimetric mapping problem through simulation and real-life imaging. However, it is also necessary to understand how commodity cameras process images via what is commonly referred to as the camera imaging pipeline.

There are many steps in the camera imaging pipeline to convert the initial sensor response to the final image outputted for viewing. Cameras have dedicated hardware known as image signal processors (ISPs) that manipulate the image in a pipeline fashion to produce an image that is typically used for human viewing. An overview of the camera pipeline (i.e., an ISP) is shown in Fig. 2.8.

The basic ISP starts with a sensor raw-RGB image that contains the values right after they have passed through the colour filter array (CFA) and are captured by the sensor: a one-channel raw image that is in mosaiced format. The first step to process the image is to perform a black level subtraction. The black level is the value recorded by the sensor if no light is received. The image is then normalized between $[0 - 1]$ using the white level value (the largest possible value captured by the camera) [29]. Usually this number depends on the bit depth of the raw image; for example a 12 bit image has a white level of $2^{12} = 4096$.

A raw-RGB image often does not have uniform light on all portions of the sensor due to lens vignetting and the angle with which light hits the sensor. A vignetting correction must be stored to brighten the lighting when moving away from the center of the image [29, 30].

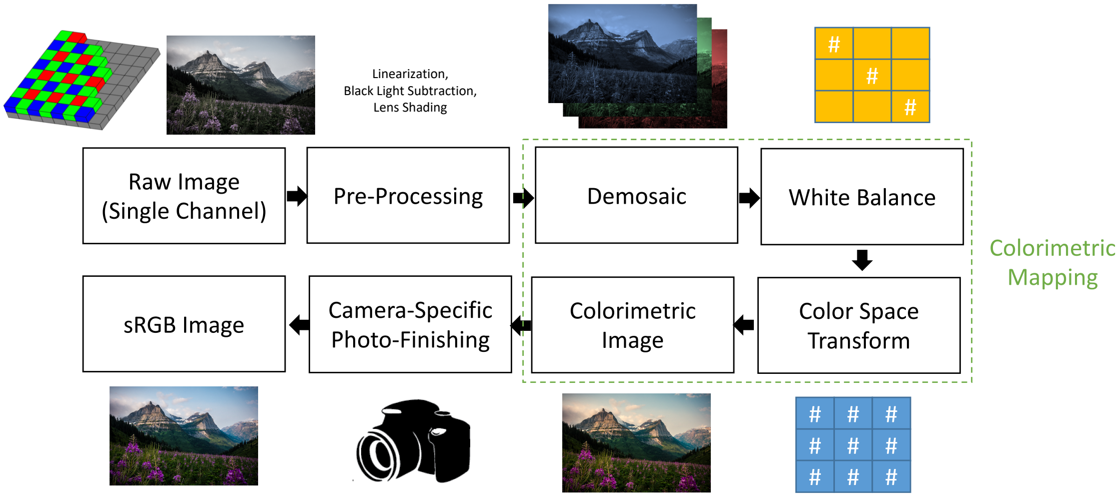


Figure 2.8: This diagram (adapted from [27]) illustrates the typical routines applied as part of a camera pipeline. Steps are detailed in the thesis. Picture of mountains is public domain [28].

Another lens effect that is always corrected is warping. The image that is captured by a lens will have distortion due to the lens properties. This distortion can be undone by using the lens properties [31] or looking for some set of regularly spaced fiducials in the scene [32].

After the lens shading and distortion corrections, this image is demosaiced depending on the pattern of the CFA, also known as Bayer filter [33]. Demosaicing is the process of estimating a three channel color image from the single-channel Bayer image. There are four common patterns used for Bayer filter: GBRG, BGGR, RGGB, GRBG [34]. The order of the letters represents the tiling of the colors (red(R), green(G), and blue(B)) from left-to-right, top-to-bottom in a 2x2 region. Notice that green occurs twice in the pattern and composes 50% of the Bayer filter. Since the human eye is more sensitive radiant energy present in the “green” wavelengths, it has become standard in cameras to have more green as a way of noise reduction [35, 36]. There are many demosaicing methods that are available. Some demosaicing methods are simpler than others, while others that are more complex focus on retaining spatial information. The simplest demosaicing algorithms include nearest neighbors [37], basic linear/cubic interpolation [38], and smooth hue transition interpolation [39, 40]. The demosaicing problem is complicated because it is essentially trying to fill in information about an image that is missing, hence sometimes algorithms can face issues with artifacts and edge preservation [41]. More complicated algorithms like adaptive color plane interpolation [42], weighted gradient based interpolation [43–45] and *a posteriori* decision algorithms [46] are used to help remove these artifacts. These algorithms are usually followed with a refine step to remove more artifacts [41].

After demosaicing, the raw image is now represented as a three-channel image. However, it is still a raw image in the camera's color space. For it to have any meaning, it must be mapped to the standard colorimetric space, CIE XYZ [47, 27]. This is a two-step process involving a white balance operation and a color space transform (CST) [48]. These collectively perform the colorimetric mapping from raw-RGB space to a device-independent space.

White balance is primarily used for two goals that are shown in Fig. 2.9: color constancy and aesthetic visual appearance. Both goals are equally valid and the choice between these two objectives depend on the problem being solved [49]. For scientific or measurement purposes, color constancy is nearly always the goal because then colors in an image represent the reflectances of objects in the scene [50]. However, consumer cameras usually attempt to impart a unique visual appearance that is visually appealing for the consumers [49].

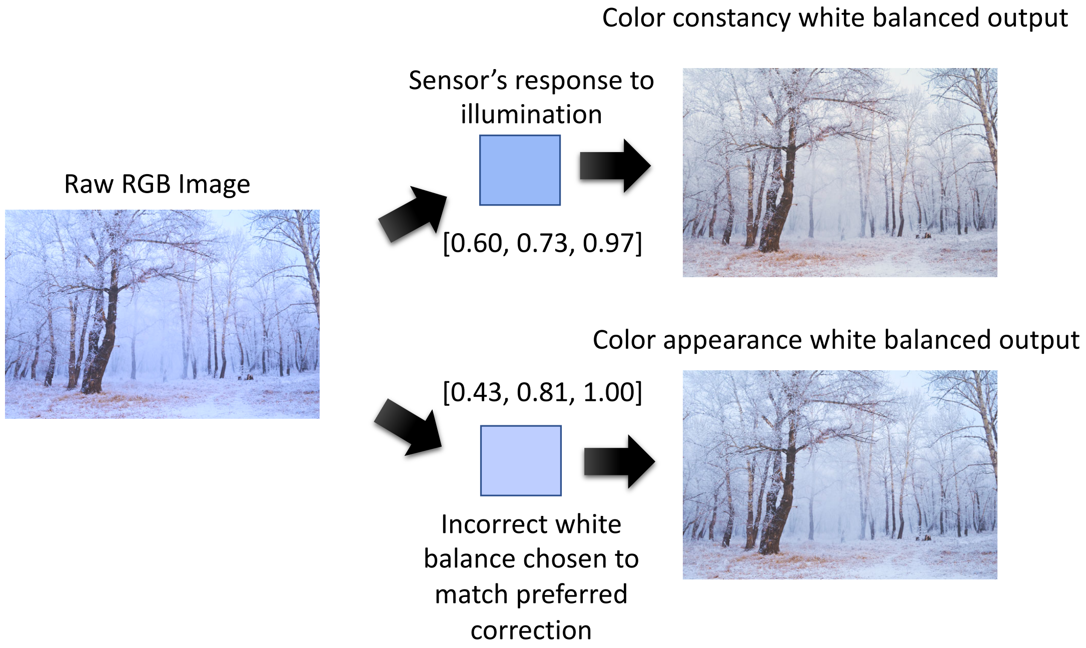

Color constancy white balanced output
Sensor's response to illumination
[0.60, 0.73, 0.97]
Raw RGB Image
Color appearance white balanced output
[0.43, 0.81, 1.00]
Incorrect white balance chosen to match preferred correction

Figure 2.9: This figure illustrates the two types of white balance. The top image is a white balance for a colorimetric purpose while the bottom image is a white balance used for a preferred correction for aesthetic reasons. Image of forest is public domain [51].

White balance for color constancy is the process of balancing the colors in the image with respect to the sensor's response to the illumination [52, 53]. This means that after an ideal white balance, the sensor will capture the same image regardless of the illumination [54]. Notice that the sensor response to the illumination can also be thought of as the response to any white object in the scene, because an ideal white reflects all wavelengths evenly. An image without white balance may have a color cast due to the illumination and also varying sensitivities between the color channels [55]. If the sensor response to the illumination is given by $[r_l, g_l, b_l]$ then a white balance operation is given by:

$$\begin{bmatrix} R_w \\ G_w \\ B_w \end{bmatrix} = \begin{bmatrix} 1/r_l & 0 & 0 \\ 0 & 1/g_l & 0 \\ 0 & 0 & 1/b_l \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where R, G, B are the input, and R_w, G_w, B_w are the colors after applying the white-balance matrix. It should be apparent that this will ensure that the sensor's response to the illumination $[r_l, g_l, b_l]$ will lie on the achromatic (white) line $R = G = B$. Thus the name "white balance."

The second type of white balance is with respect to color appearance. This type of white balance is used as part of photo-finishing for aesthetic purposes [49]. This could be interpreted as an incorrect white balance because the output image may have a color cast. Lower color temperatures result in a red-yellow color cast while higher color temperatures results in blue color casts. When the goal of white balancing is color appearance, the image will be white balanced to maintain visually pleasing color casts. However, a color constancy approach would try to make sure that this white object would always appear as the same color regardless of the lighting condition [52, 53].

It is worth noting that computing an accurate white balance correction for a given scene is a challenging problem because it is difficult to know what the response to illumination for a scene is [56]. Manual white balance allows the user to select the appropriate correction. The diagonal correction matrix is calibrated in the factory for various illuminations. The user selects which of the calibrated white balance matrices to use for the given scene. However, for applications that plan to use the camera as a consumer measurement device in the real world it will almost be done with auto white balance (AWB). There are many AWB algorithms in the literature. A discussion on AWB is outside the scope of this thesis, however, a good survey can be found here [56]. Choosing a wrong white balance can cause an incorrect interpretation of color; colorimetric data can easily be lost in this step [57]. As a result, in a real setting, it will be necessary to use a color calibration pattern to help establish the correct white-balance. This will be described in more detail in the Sec. 3.

On an ISP, the color space transform (CST) matrix is estimated by using an interpolation between two factory color correction matrices calibrated to two different color temperatures. The color temperature of the white balance matrix is used to interpolate the two factory correction matrices. This method for correcting scenes covers most general lighting and is a simple method to approach for correcting color in all the various lighting sources that are encountered by a camera. However, this is not perfect because it assumes all lighting exists exactly between two color temperatures; real world lighting always has slight anomalies. Recently, researchers have been trying to improve this step by adding a third factory correction matrix at a color temperature in the middle of the other two matrices [20]. What is most important to take away from this part of the ISP is that this matrix is not perfect for all illuminations and is just an estimate depending on the scene content. Another factor that makes this worse is that the white balance temperature dictates the CST matrix. This means if an incorrect white balance is chosen, which is highly likely [54], our colorimetric errors wil increase in the CST step as well.

After the image is multiplied by the white balancing matrix and CST matrix, the image is now a 3-channel CIE XYZ image. However, the standard RGB (sRGB) color space is used by most devices to display images. There is a fixed transformation from XYZ to

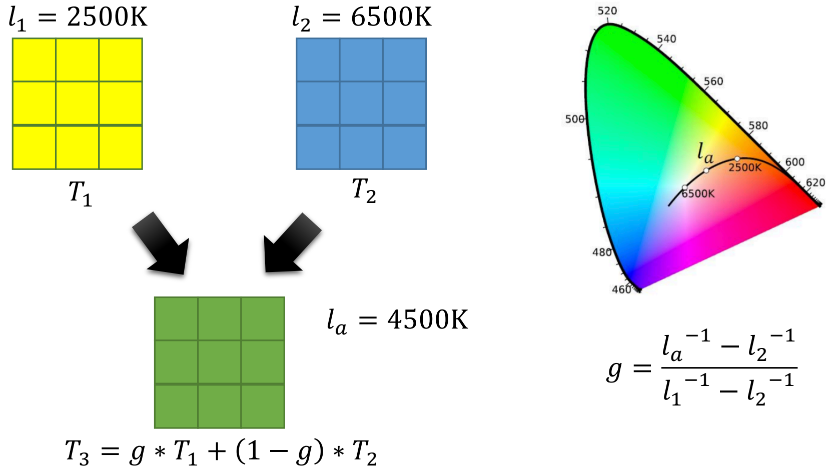


Figure 2.10: This figure illustrates how the color space transform is generated on most cameras. The yellow and blue 3×3 boxes represent the coefficients of a 3×3 matrices, T_1 and T_2, that represent the color space mappping matrices. These matrices are precomputed for the two different illuminations l_1 and l_2 that have color temperatures 2500K and 6500K. To compute the a new color mapping T_3 (green 3×3 boxes) for an illumination l_a, a weighting is computed. An intermediate value g is calculated that represents the weighting of T_1 and T_2 that is used to calculate T_3. Image of color gamut taken with permission from [25].

sRGB that is used to get the data in sRGB format.

Most cameras have additional photo-finishing steps, including color manipulation and tone mapping, that further change an image so that it is closer to the human perception of that scene [58, 48]. These additional steps remove colorimetric information because the aim of these steps is to create aesthetic images [58]. For pipelines that want to use cameras as a measurement device, they should be ignored.

After these camera-specific photo-finishing techniques, data is transformed from XYZ to linear sRGB, a non-linear perceptual gamma is applied [27]. This gamma essentially scales the brightness of colors so they correspond to the non-linear perception of brightness of the human visual system [59]. Humans are more sensitive to changes in darker colors than they are to lighter color; a gamma allows these darker colors to be stored at a higher perceptual resolution [60, 61]. The gamma encoded into sRGB must be decoded by the monitor or viewing device [61]. Thus, a successful use of gamma will not change the color of the image to a human viewing it.

It is worth noting that the CIE XYZ space can be converted to CIE $L^*a^*b^*$ (or CIE Lab for short). CIE Lab is a color space that is used for understanding perceptual differences in colors. This space is needed because CIE XYZ does not uniformly encode colors with respect to perceived color differences. For example, a small change in CIE XYZ values in the bluish colors will be perceived more than a similar change to green colors. The space Lab is organized with three axis corresponding to lightness/darkness (L), red/green(a), and blue/yellow (b). The delta E metric is a perceptual difference metric that uses CIE Lab values. Initially, this delta E metric was defined as Euclidean distance in Lab space. In 2000, the metric was refined because the CIE Lab space was found not to be as perceptually uniform as first thought. The main corrections were hue rotations for certain blues and compensation for neutral colors, lightness, chroma, and hue [62].

For the work in this thesis, we will provide error reporting in both a percent error related to distance in XYZ and a delta E error which represents a perceptual error in CIE Lab space. A scientific measurement company may be more interested in XYZ errors. However, an entity creating images for human viewing might be more interested in the perceptual error. Both errors have their place in the discussion and we decide to report both numbers in this thesis.

Chapter 3

Approach

As discussed in Chapter 1, the goal of the work in this thesis is to determine how prevalent color imaging failure is in a real-world setting. Chapter 2 described the two major components at play: the sensor-specific spectral response and the spectral profile of the scene's illumination. With this in mind, we are particularly interested in determining if there are pairs of sensor-illumination that result in a colorimetric mapping with a error that would be considered a failure for a given task. When colorimetric failure does occur, we are interested in determining if this is attributed mainly to the sensor or to the scene illumination.

To investigate color imaging failure, we conduct two experiments. Our first experiment is a simulation that synthesizes color chart images using databases of illuminations, reflectances, and camera sensitivities. The second experiment works directly with real camera sensors, scene illumination, and a calibration chart. For this second experiment, it is necessary to apply some of the camera pipeline steps described in Chapter 2.

3.1 Synthetic image experiments

Fig. 3.1 provides an overview of our synthetic experimental design. We simulate the capture of raw-RGB sensors images composed of different camera sensor and source light pairs. Specifically, we construct a database of the spectral sensitivity of 28 camera sensors estimated by Jiang et al. [63] and 69 camera sensors provided by Image Engineering [64]. Note that these datasets store spectral information from 400-720nm at step sizes of 10nm and 5nm, respectively. We bi-linearly interpolate both datasets to step sizes of 4nm. This re-sampling is performed to match the sampling of some of the illumination spectral data as will be discussed shortly. This gives us a total of 97 spectral sensitivity profiles representing different consumer-grade sensors, including DSLR, point-and-shoot, and smartphone cameras. As shown in Fig. 3.1, we represent a specific camera, S, spectral sensitivity as a $3 \times N$ matrix, $\mathbf{C}^{\mathbf{S}}$, where N is the number of spectral samples over the visible spectrum (i.e., 400-720nm) and the rows of $\mathbf{C}^{\mathbf{S}} = [cR; cG; cB]$ correspond to the R, G, B channels.

Our database of lighting spectra is composed from two sources. First, we use the data set from Barnard et al. [21] that contains 102 spectral power distributions of commercial lighting at a step size of 10nm. While this is an older data set, it provides a comprehensive data set of common commercial and environment lighting. We augmented this data set with recent light sources purchased at a home supply shop—namely a daylight D65 light, a standard fluorescent light, an incandescent light, and an adjustable RGB LED light (Philips Hue) that is intended for mood lighting. For the adjustable LED lighting, we tune it to its three extremes—maximum red, green, and blue. We measure the spectral profile of these added lights using a UPRtek PG200N PAR meter with a stepsize of 4nm. As done with

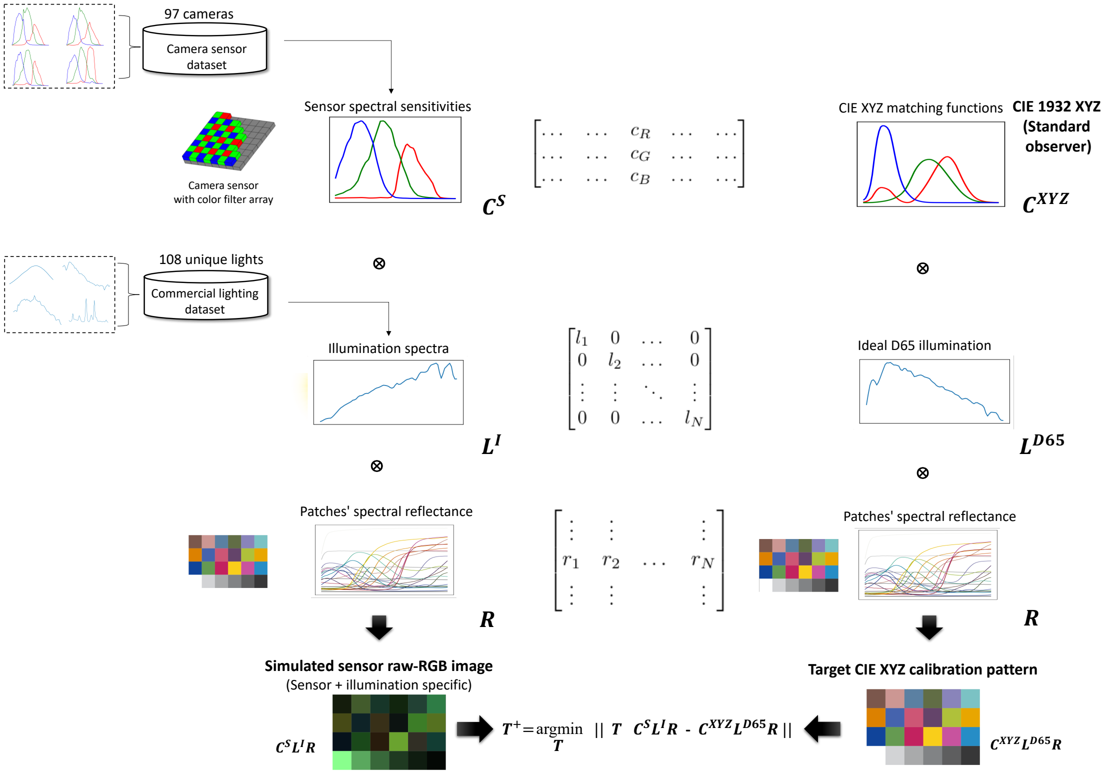

97 cameras
Camera sensor dataset
Camera sensor with color filter array
Sensor spectral sensitivities
C^S
CIE XYZ matching functions
CIE 1932 XYZ (Standard observer)
C^{XYZ}
⊗
108 unique lights
Commercial lighting dataset
Illumination spectra
L^I
Ideal D65 illumination
L^{D65}
⊗
Patches' spectral reflectance
R
Patches' spectral reflectance
R
Simulated sensor raw-RGB image
(Sensor + illumination specific)
$C^S L^I R$
Target CIE XYZ calibration pattern
$C^{XYZ} L^{D65} R$
$T^{+} = \underset{T}{\text{argmin}} \ || \ T \ C^S L^I R \ - \ C^{XYZ} L^{D65} R \ ||$

Figure 3.1: This figure illustrates our analysis using synthetically generated raw-RGB images. A database of sensors and illuminations is used to emulate different raw-RGB combinations. For each raw-RGB image, we estimate a 3×3 linear colorimetric mapping functions, $\mathbf{T}^{+}$, that maps the raw-RGB values to the target colorimetric values. The residual error of this mapping is used to gauge if the color mapping has failed.

the camera sentititvy data, the data from Barnard et al. [21] is bi-linearly re-sampled to a step size of 4nm. As shown in Fig. 3.1, we represent each lighting condition, denoted as $\mathbf{I}$, as a $N \times N$ diagonal matrix, $\mathbf{L^I}$, where each element in the diagonal is the spectral sample $1..N$ over the visible spectrum.

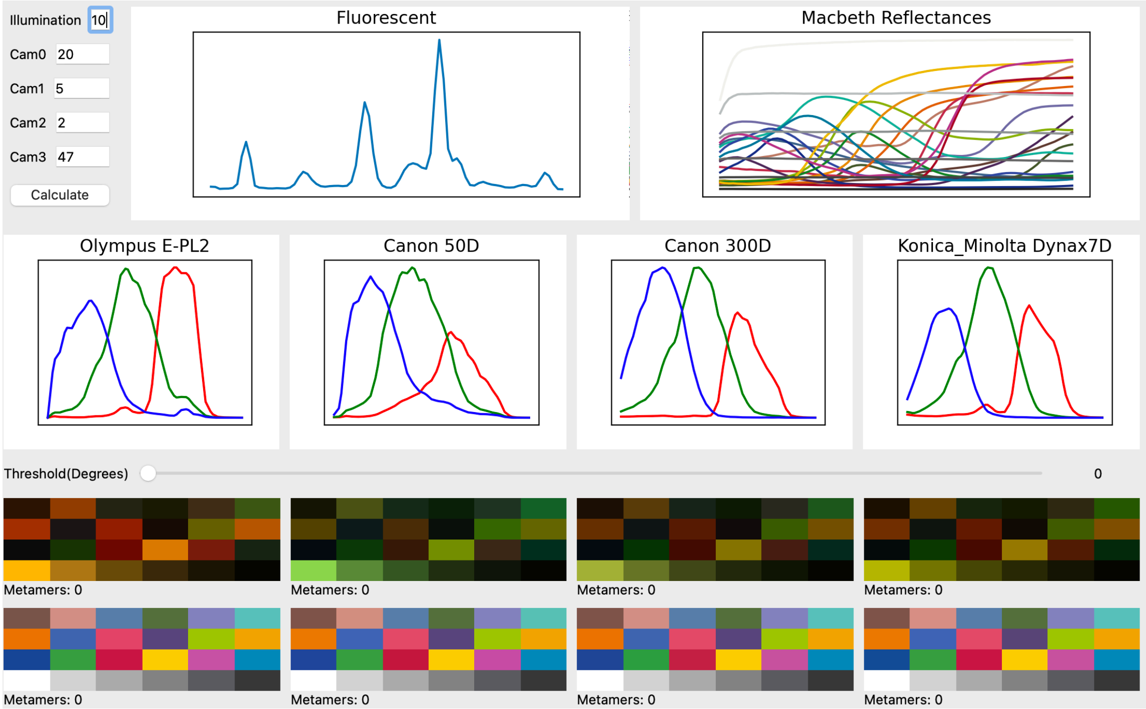
For the color calibration pattern, we use the standard 24-patch X-Rite color rendition chart, specifically the bottom 24 colors of an X-Rite Passport [65]. We also run our experiments with the 140-patch X-Rite ColorChecker Digital SG [65] but noticed no significant improvements. For the purposes of this thesis, we opted to use the more common 24-patch color chart. The spectral reflectance of each color patch was measured by Chomaxion with step size of 10nm; we use the Chromaxion data stored under the entry "ColorChecker Passport #2" [65]. All patches are combined into a single $N \times 24$ matrix, $\mathbf{R}$, where each column of the matrix represents a patch's spectral reflectance coefficients between $[0 - 1]$ for each spectral sample $1 \ldots N$. For all datasets, our spectral data is represented by $N = 81$ bands from 40nm to 720nm with a step size of 4nm. For datasets not captured with 4nm resolution, we use bi-linear interpolation to achieve a 4nm step size.

By sampling camera sensors and light sources from the datasets, we can synthesize a raw-RGB image for a specific camera sensor S and lighting condition I combination through matrix multiplication as follows

$$\text{raw-RGB image} = \mathbf{C^S} \cdot \mathbf{L^I} \cdot \mathbf{R}, \tag{3.1}$$

where $\cdot$ represents matrix multiplication. Note that each S and I pair will result in 24 raw-RGB values corresponding to the sensor's observation of a color calibration patch under the given lighting condition.

The simulation framework includes a visual interface that shows the camera sensitivities, illumination spectra, and reflectance information. This simulation interface is shown in Fig. 3.2. The simulation interface contains our database of lighting spectra and camera sensitivities. It allows us to view how any light condition is synthesized in raw for any four cameras (last to bottom row of color patches). Then the bottom row of color patches show the colors after a color correction matrix $\mathbf{T}^{+}$ is calculated and applied for each camera.



Illumination 10

Cam0 20

Cam1 5

Cam2 2

Cam3 47

Threshold(Degrees) 0

Metamers: 0

Metamers: 0

Metamers: 0

Metamers: 0

Metamers: 0

Metamers: 0

Metamers: 0

Metamers: 0

Figure 3.2: This figure illustrates our PyQT interface that contains the simulation framework. The simulation framework includes control over the illumination and camera while having fixed reflectances. These reflectances are the standard MacBeth reflectances.

The calibration chart CIE XYZ values are computed as follows:

$$\text{target XYZ} = \mathbf{C^{XYZ}} \cdot \mathbf{L^{D65}} \cdot \mathbf{R}, \tag{3.2}$$

where $\mathbf{C^{XYZ}}$ is the CIE 1932 XYZ spectral matching functions, $\mathbf{L^{D65}}$ is a matrix with the CIE Standard Illuminant D65 spectral values.

Our simulation mimics a well-lit environment with proper exposure and focus settings. Based on the synthesized data, we perform colorimetric calibration by computing a 3×3 linear transform, $\mathbf{T}^{+}$, between the synthesized raw-RGB image and the target XYZ as follows:

$$\mathbf{T}^{+} = \underset{T}{\operatorname{argmin}} ||\mathbf{T}\mathbf{C^{S}} \cdot \mathbf{L^{I}} \cdot \mathbf{R} - \mathbf{C^{XYZ}} \cdot \mathbf{L^{D65}} \cdot \mathbf{R}||, \tag{3.3}$$

where $\mathbf{T}^{+}$ is computed using standard pseudo-inverse methods. We note that there are more sophisticated mapping functions (e.g., [9, 10]); however, a 3×3 transform remains a standard choice for colorimetric calibration [6] and is sufficient for our analysis.

3.2 Real-images experiments

Our second experiment focuses on real images captured from smartphones using standard lighting found in a home environment. Fig. 3.3 shows an example of our setup to capture input images from smartphones. A camera is placed in a lightbox that is illuminated with the lights mentioned above in Sec. 3.1 from a home supply shop. We tested five smartphones from three major smartphone providers: Google Pixel 5, Samsung Galaxy S8, Samsung Note 10, Apple iPhone 8, and Apple iPhone 12. A custom calibration chart

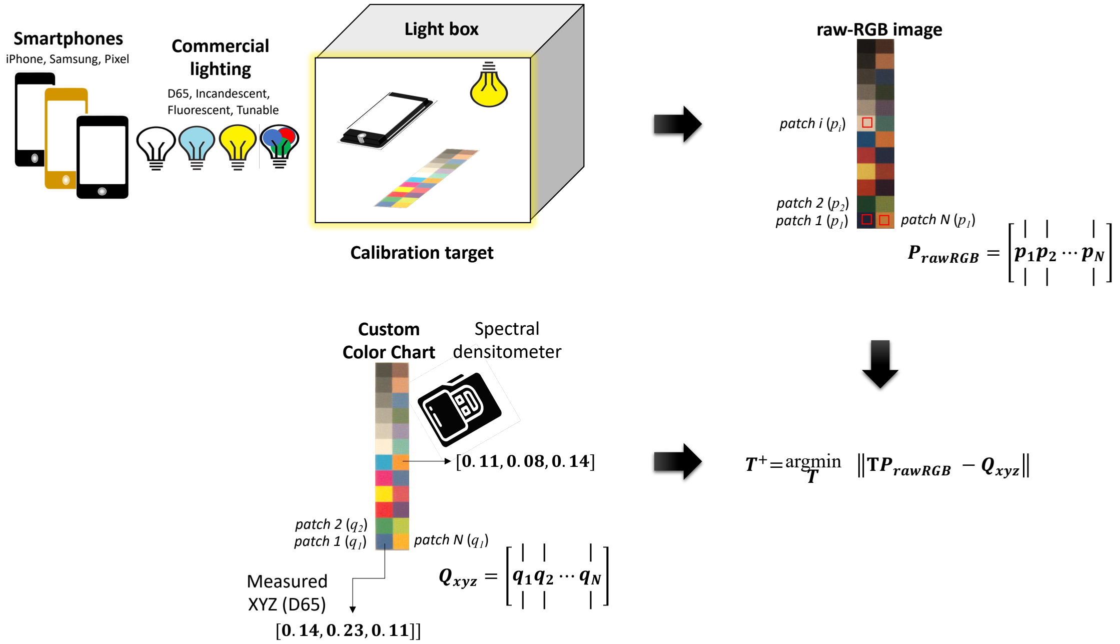


Figure 3.3: This figure illustrates our analysis using images captured with smartphone devices. The cameras are placed in a light box and are used to capture an image of a custom color calibration pattern or X-Rite Passport. The calibration pattern is captured under several different commercial lights, including a tunable LED light. The patches on the calibration pattern have had their CIE XYZ values measured using a spectro-densitometer. Working directly from the camera's raw-RGB image, a 3×3 color mapping $\mathbf{T}^{+}$ is computed based on patch correspondences as shown.

was designed with 24 color patches printed to mimic the X-Rite chart. The color patches were measured with an Techkon spectro-densitometer to obtain their CIE XYZ values under a target illumination (D65). We represent each of these measured color patches as a 3×1 XYZ vector denoted as $\mathbf{q}_i$. While the custom calibration chart's color gamut is smaller than the X-Rite, this low-cost printed color chart is indicative of what would be used in a home-based diagnostics setting.

The camera is placed in a lightbox where multiple bulbs of the same type are used to ensure a uniform illumination. A lux meter is placed at the bottom of the lightbox to allow us to measure the intensity of the light. We then use a dimmer to adjust the intensity of the bulbs so we can capture different lighting conditions under the same lux. Since, the max range of the LED lights individually was only 100 lux, we also took images with all the other bulbs (D65, incandescent, and fluorescent) at this lower light level. However, to ensure that corrections worked under well-lit conditions, we also capture data for D65, incandescent, and fluorescent lights at 500 lux.

We capture images of the color chart in "RAW" mode on the device. On android devices, we capture raw images with the Open Camera app. We collect Apple iPhone raw-RGB images with our own custom app that allows the images to be captured in raw format since that feature is not natively available in the iPhone camera app; this could also be replaced with other alternatives that are freely available. We capture all images with automatic exposure and automatic focus; this is done to emulate how users would normally use their personal devices. The sensor's raw-RGB values are represented as a 3×1 XYZ vector denoted as $\mathbf{p}_i$. Similar to the simulated imaging experiments, we perform colorimetric calibration by computing a 3×3 linear transform, $\mathbf{T}^{+}$, between the real camera raw-RGB image and the target XYZ as follows:

$$\mathbf{T}^{+} = \underset{T}{\mathrm{argmin}}||\mathbf{T}\mathbf{P}_{\mathbf{rawRGB}}{}^{\mathbf{L}} \cdot \mathbf{L}^{\mathbf{I}} \cdot \mathbf{R} - \mathbf{Q}_{\mathbf{XYZ}}||, \tag{3.4}$$

where $\mathbf{L}$ represents the different illumination, $\mathbf{P}_{\mathbf{rawRGB}}{}^{\mathbf{L}}$ is a matrix composed of the 24 vectors $\mathbf{p_i}$ under illumination $\mathbf{L}$, and $\mathbf{Q}_{\mathbf{XYZ}}$ is a matrix composed of the corresponding 24 vectors $\mathbf{q_i}$ of the custom color chart. Again, $\mathbf{T}^{+}$ is computed using standard pseudo-inverse methods.

Working with real images is more complicated than synthetic images because raw-RGB images have to be processed, at least partially, via the ISP pipeline described in Sec. 2.2. Raw images are normally stored in DNG (Digital Negative) format [66]; the DNG metadata must be used for removing lens shading (vignetting) effects, demosaicing, and applying white-balance. The DNG specification stores some of its metadata in tags and other parts in opcodes. The data from tags describes some of the parameters we use include black level, white level, CFA pattern, and white-balance. Opcodes describe parameters and transformations that need to be applied to a raw image; the transformations specified by opcodes change from camera to camera. We emulate the camera pipeline by using the Simple Camera Pipeline [67] to process most of these tags.

This pipeline performs (1) black level subtraction, (2) image normalization by using the white level value, and (3) demosaicing as described in Chapter 2. We also use the white-balance vector in the DNG containing three values that represents the relative intensities between the red, blue, and green channel. It is important to note that estimation of the color mapping function $\mathbf{T}^{+}$ is able to implicitly perform the required white-balance operation. However, we still include the white balancing because it is standard procedure when working with RAW images.

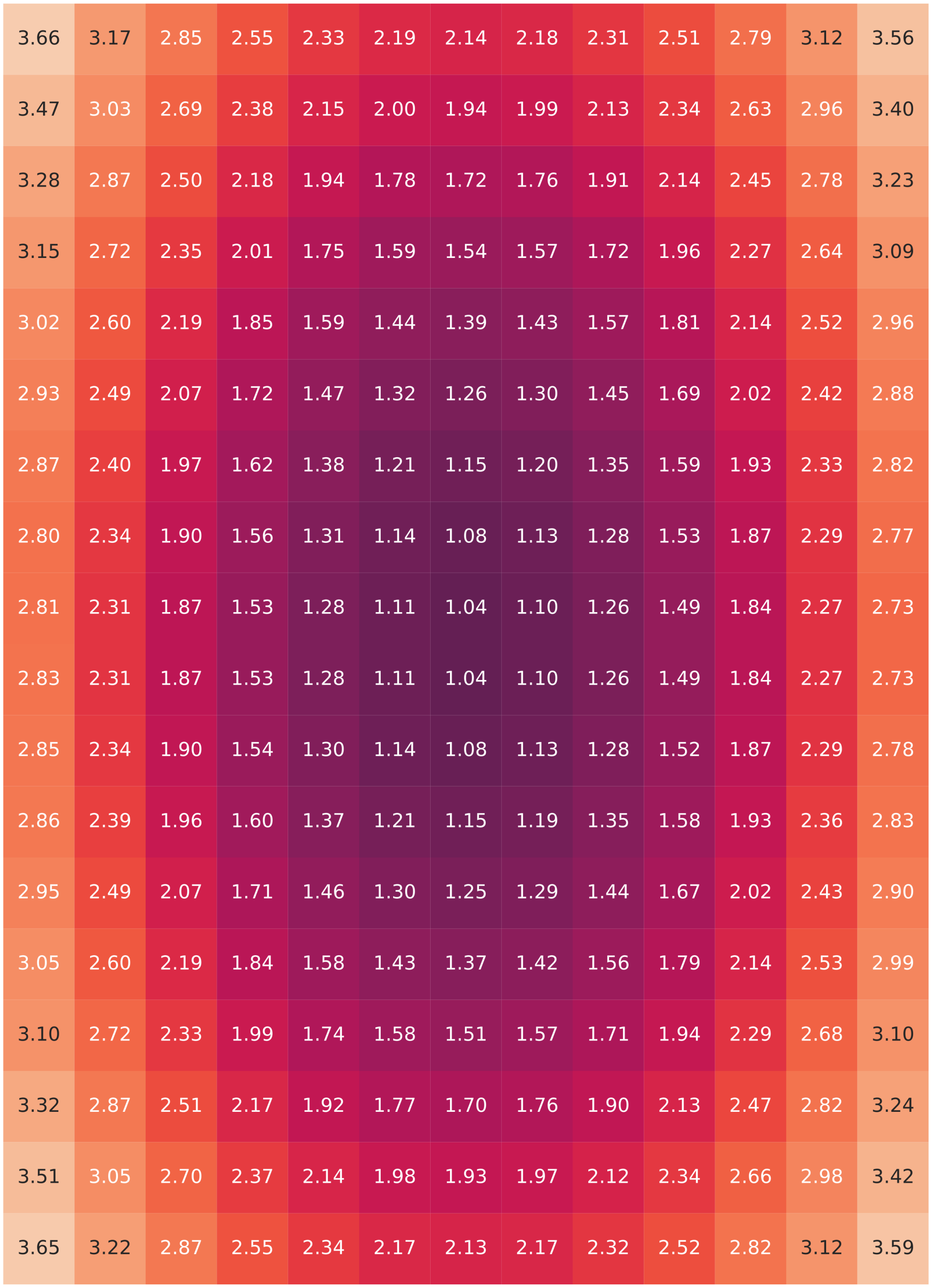
It is also worth noting that we encountered some difficulty when processing images from iPhones. We noticed that iPhones stored lens shading information in different DNG opcodes than from Android phones. Specificaly, iPhones store lens shading information by specifying a radial fall off using five parameters, k_{0-4}, and another two parameters, c_x, c_y, to store the optical center of the image. The gain g that needs to be applied for a certain pixel is specified in terms of r, the normalized Euclidean distance of the optical center from the pixel, and is computed as follows:

$$g = 1 + k_0 * r^2 + k_1 * r^4 + k_2 * r^6 + k_3 * r^8 + k_4 * r^{10}.$$

In contrast, Android stores this same lens shading information in a lens shading map. This map is a lower resolution image of the full image but it spatially describes the light fall off in the image. An example of an Android lens shading map is shown in Fig. 3.4. An iPhone gain map would look very similar to this map.

The functionality to process an Android lens shading map is already present in the Simple Camera Pipeline [67], but the ability to use the iPhone radial fall off for lens shading is not available. As a result, we create a function to compensate for lens shading on iPhone. Then, we add our own code to treat iPhone and Android images different, based off the opcodes present. This gives us a fully functioning camera pipeline for both iPhone and Android.

After capturing an image of the calibration chart with the smartphone cameras, we have the problem of locating the color calibration chart. There are methods to find and locate these charts using segmentation [68] but these algorithms are not trivially extendable to our



3.66	3.17	2.85	2.55	2.33	2.19	2.14	2.18	2.31	2.51	2.79	3.12	3.56
3.47	3.03	2.69	2.38	2.15	2.00	1.94	1.99	2.13	2.34	2.63	2.96	3.40
3.28	2.87	2.50	2.18	1.94	1.78	1.72	1.76	1.91	2.14	2.45	2.78	3.23
3.15	2.72	2.35	2.01	1.75	1.59	1.54	1.57	1.72	1.96	2.27	2.64	3.09
3.02	2.60	2.19	1.85	1.59	1.44	1.39	1.43	1.57	1.81	2.14	2.52	2.96
2.93	2.49	2.07	1.72	1.47	1.32	1.26	1.30	1.45	1.69	2.02	2.42	2.88
2.87	2.40	1.97	1.62	1.38	1.21	1.15	1.20	1.35	1.59	1.93	2.33	2.82
2.80	2.34	1.90	1.56	1.31	1.14	1.08	1.13	1.28	1.53	1.87	2.29	2.77
2.81	2.31	1.87	1.53	1.28	1.11	1.04	1.10	1.26	1.49	1.84	2.27	2.73
2.83	2.31	1.87	1.53	1.28	1.11	1.04	1.10	1.26	1.49	1.84	2.27	2.73
2.85	2.34	1.90	1.54	1.30	1.14	1.08	1.13	1.28	1.52	1.87	2.29	2.78
2.86	2.39	1.96	1.60	1.37	1.21	1.15	1.19	1.35	1.58	1.93	2.36	2.83
2.95	2.49	2.07	1.71	1.46	1.30	1.25	1.29	1.44	1.67	2.02	2.43	2.90
3.05	2.60	2.19	1.84	1.58	1.43	1.37	1.42	1.56	1.79	2.14	2.53	2.99
3.10	2.72	2.33	1.99	1.74	1.58	1.51	1.57	1.71	1.94	2.29	2.68	3.10
3.32	2.87	2.51	2.17	1.92	1.77	1.70	1.76	1.90	2.13	2.47	2.82	3.24
3.51	3.05	2.70	2.37	2.14	1.98	1.93	1.97	2.12	2.34	2.66	2.98	3.42
3.65	3.22	2.87	2.55	2.34	2.17	2.13	2.17	2.32	2.52	2.82	3.12	3.59

Figure 3.4: This figure shows a lens shading map taken from a Pixel 5. For display purposes, this lens shading map was downsampled to 12x16.

custom color chart. Initially, we ran color chart detection code to find the X-Rite chart, but we found that in bad lighting conditions (e.g., the blue LED lighting) the chart sometimes would not be found in the image. Instead, we capture the charts by pasting them on top of a small card with Aruco Markers [69]. The Aruco markers are fiducials that resemble QR codes. We print four aruco markers onto each calibration chart.

We use the OpenCV library to detect the fiducials, unwarp the image, and crop the image to just the portion we were interested in. Then, we scale all these cropped card images to the same size. Fig. 3.5 shows examples of cropped images for both color charts.

Once the color calibration portion of the image has been cropped and unwarped, we use the fixed color chart patch layout to retrieve the colors of each color patch. For noise reduction, we sample a 10×10 pixel area of each of the 24 colors and take the median.

Another interesting problem we ran into is that identifying fiducials in raw-RGB images is not always reliable with the OpenCV library. In particular, OpenCV has been optimized for sRGB images. Linear raw-RGB images often result in poor performance. To address this, we process each raw-RGB image to create an sRGB image using the rawpy library. We pass this sRGB image to OpenCV to detect the fiducials. Since the spatial locations are identical to the raw-RGB image, we use use the results from the sRGB image to crop and warp the raw-RGB image and then discard the sRGB image. An example of the sRGB and the raw image for both color charts are shown in Fig. 3.5.

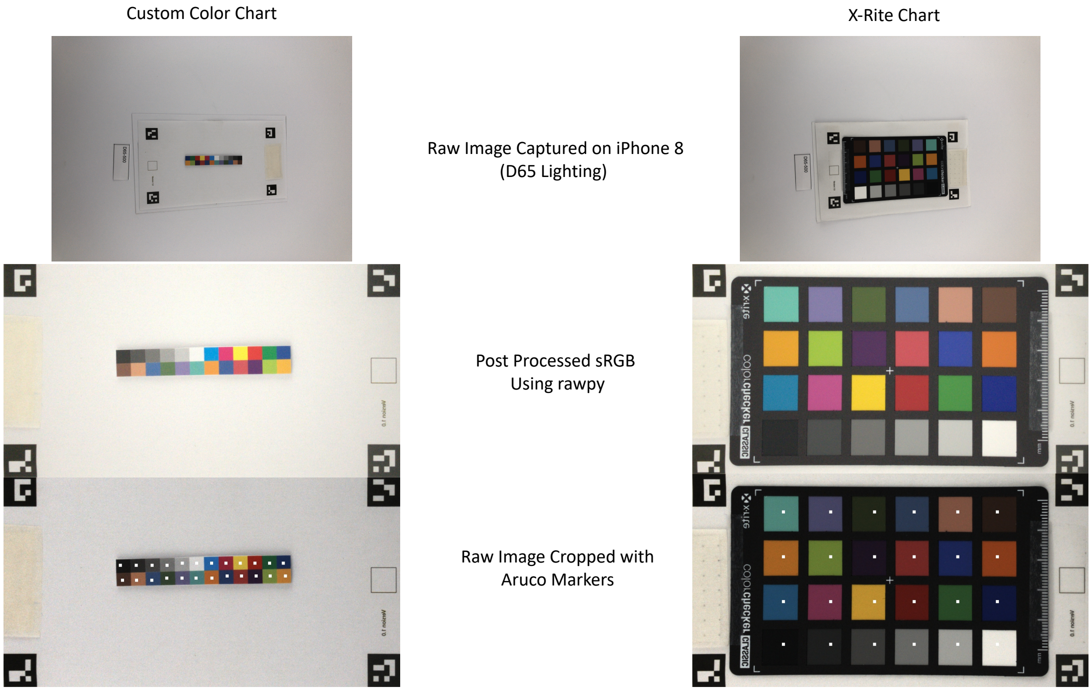


Figure 3.5: This figure illustrates the steps to find the card with the color chart on it. A sRGB image is generated first so the aruco markers can be found by OpenCV. Then the locations of the aruco markers are used to crop the card and remove lens distortion effects. In this figure, we show the card extraction for both the custom chart and X-Rite chart. This process is useful because it means the charts are in the same location for all images.

3.3 Summary

This chapter provided details to the experimental designs for color failure analysis: synthetically generated images and smartphone captured images. The following chapter analyzes the findings from these experiments.

Chapter 4

Results

In this section, we describe the findings from the experiments described in Chapter 3.

4.1 Simulated images

Our simulated experiments use 97 cameras and 108 lighting conditions, resulting in 10476 raw-RGB images for all sensor-lighting pairs in our database. Fig. 4.1 shows a histogram of all sensor-lighting pairs in terms of their calibration error plotted as a percentage; 0% error implies a perfect calibration. The vertical axis counts all sensor-image pairs that produced a particular error. The horizontal axis of the plot is the percent error, which is capped at 20% (i.e., errors above 20% are grouped together). While the exact error threshold that would be deemed color calibration "failure" would be application specific, our histogram reveals three distinct modes. The first has less than 8% error and we classify them as "good". We have 10185 image pairs fall into the "good" category. This category represented *all* sensor and lighting conditions, expect the three extremes lighting conditions obtained from the tunable light. This indicates that not a single sensor failed under any of the standard lighting from the 102 lights from Barnard et al. [21] and our additional three lights (other than the tunable light). Interestingly, there was not a single degenerate sensor-lighting pair that resulted in color calibration failure.

The second mode in our histogram is attributed to the tunable light, tuned to the maximum green. We classify these results as “poor” calibration. The spectral distribution for this green light is shown in Fig. 4.1 and is fairly broad. This impacted *all* sensors, resulting in a total of 97 images in this category. Finally, the last mode in our histogram was attributed to the blue and red lighting condition from the tunable light and resulted in the “failure” cases. This is not surprising as the spectral profiles for these lights reveal narrow band lighting. Again, *all* sensors were impacted, resulting in 194 images in this category. Fig. 1.1 in Chap. 1 shows a visual example of the results from the “good” and “failure” category.

We produce another histogram in Fig. 4.2 that represents the calibration error in terms of Delta E 2000. The Delta E error represents a perceptual error correlated with human vision. We observe similar modes in the histogram as with the fit error, except that the green LED lighting is now slightly closer towards the “good” calibration errors(less than 5). Of the 97 cameras analyzed observing a color chart in green LED lighting, 21 of the cameras showed an average Delta E error between 4-5; the remaining cameras had error from 5-8. This means although the lighting condition is not as good, the perceptual error could be adequate for certain use cases. We categorize the bad lighting condition errors as having Delta E greater than 13.

Fig. 4.2 also shows the average error for good lighting condition on individual patches.

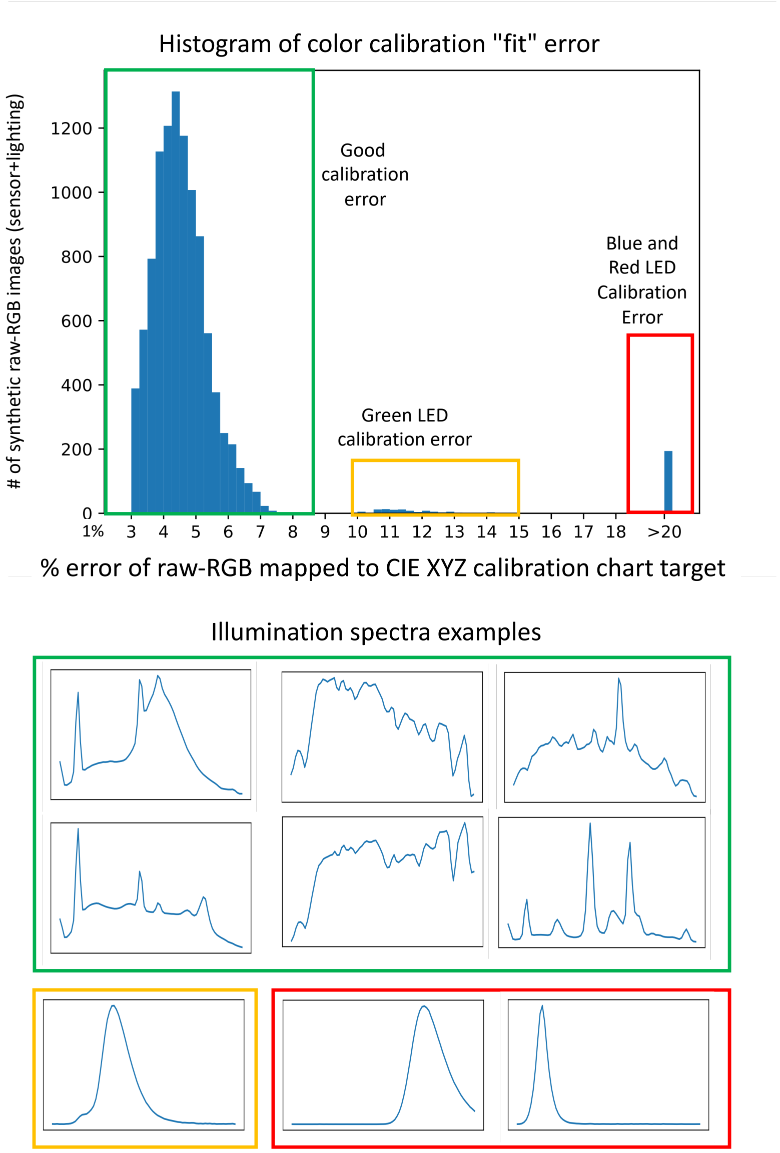


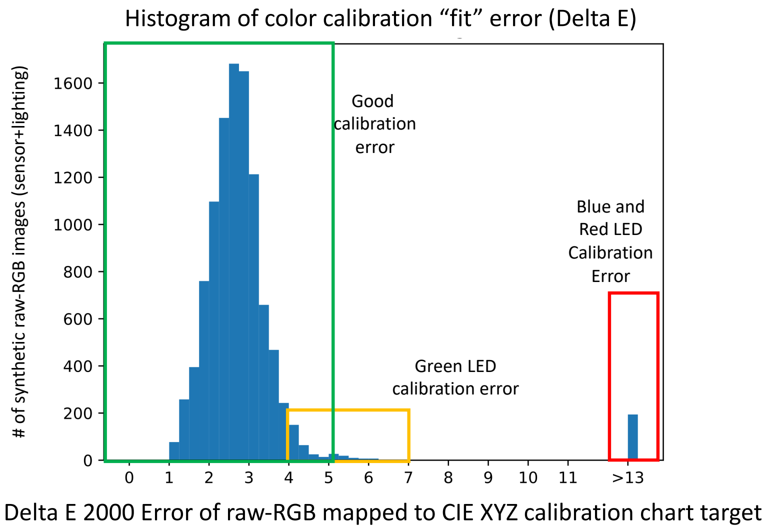
Figure 4.1: This figure shows a histogram of the synthetic raw-RGB for different sensor-lighting pairs plotted against the fit error (in percentage) to the target CIE XYZ color values. The figure shows that the vast majority of images result in a color calibration error that would be categorized as a "good" (less than 8%) error. Errors between 10–13% are considered "poor" calibration, and those above 19% are considered "failures". Out of all 108 lights, the "poor" and "failure" images are attributed to the lighting and not the sensor. As shown in the figure, the lighting for these failure cases have limited spectra.

Table 4.1: Calibration errors (in terms of % fit error) between using a 24 patch chart and a 140 chart.

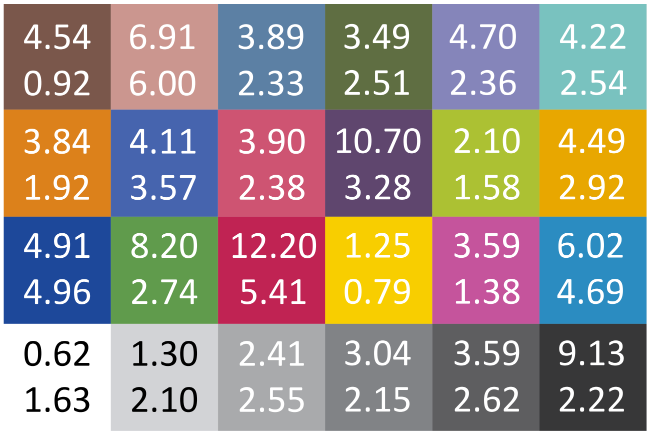
Number of Patches	Normal	G-LED	R-LED	B-LED
24	4.90	12.17	41.92	42.01
140	5.28	21.55	62.86	75.31

We see that there is correlation between Delta E and percent error. However, if we look at patch two (light skin) versus the patch 14 (green), we see that although green has a higher percent error it results in a lower Delta E error. The lower Delta E is attributed to the non-uniform nature of our visual response to different colors.

In all our experiments, we assume a 24 patch color checker; however, we could have used a color checker with more patches. We tested if more patches would create a better correction, specifically for LED lights, by redoing the experiments with the ColorChecker Digital SG 2005 (data from Chromaxion) using all 140 patches. However, there were no improved results. With just the 24 patches on an X-Rite Chart for normal lighting conditions there was an average fit error of 4.90%. This increased slightly to 5.28% when using the 140-patch color checker. Table 4.3 shows the errors between using 24 patch and 140 patch color checkers. The reason the errors don't improve for the 140 patch color checker is that the errors are "balanced" out. Some of the new patches in the 140 patch color checker provide more spectral information because their reflectances provide information in the spectral range the LED lighting produces. However, many of the other new patches provide no additional information due to the reflectances not corresponding to the LED light spectrum. The least squares model has to reconcile both of these types of patches and thus the errors do not get better.


Histogram of color calibration “fit” error (Delta E)

%Error (Top) Delta E Error (Bot) – Avg Per Color in Good Lighting



4.54 0.92	6.91 6.00	3.89 2.33	3.49 2.51	4.70 2.36	4.22 2.54
3.84 1.92	4.11 3.57	3.90 2.38	10.70 3.28	2.10 1.58	4.49 2.92
4.91 4.96	8.20 2.74	12.20 5.41	1.25 0.79	3.59 1.38	6.02 4.69
0.62 1.63	1.30 2.10	2.41 2.55	3.04 2.15	3.59 2.62	9.13 2.22

Figure 4.2: This figure shows a histogram of the synthetic raw-RGB for different sensor-lighting pairs plotted against the Delta E 2000 to the target CIE XYZ color values. The figure shows that the vast majority of images result in a color calibration error that would be categorized as a “good” (less than 5) error. Errors between 5–9 are considered “poor” calibration, and those above 13 are considered “failures”. Out of all 108 lights, the “poor” and “failure” images are attributed to the lighting and not the sensor. On the bottom part of the figure, we show the average error for each color patch in good lighting across all 97 cameras. The top number on each patch is the average percent error and the bottom number is the average Delta E 2000 error.

4.2 Real images

As described in Chapter 3, our experiments on real images used a small number of cameras (Apple iPhone 8 and 12, Samsung S8 and Note 10, Google Pixel 5). A color chart was imaged by each camera under six lighting conditions, D65, incandescent, fluorescent, and a tunable RGB LED set to its three primaries (red, green, blue at 100 lux). We use two different color charts for this analysis and our results are separated depending on the chart used.

4.2.1 Custom Calibration Chart

We first investigate the errors using a custom calibration chart that has known XYZ values measured by a densitometer. Due to the the small number of testing conditions used, we do not provide a histogram, but instead show the results in terms of calibration error in Tables 4.2 and 4.3. While this is a much smaller sampling of sensors and light sources, the results reveal that good calibration is obtained under standard lighting (less than 8.5% error or 5 Delta E). Only the extreme lights from the tunable bulb result in high error that would be considered poor (more then 10% error or 5 Delta E) or failures (over 20% error or 13 Delta E). Fig. 4.4 shows a visual example for the raw-RGB inputs for the Samsung S8 and iPhone 12 under good and poor lighting. The color calibration under the spectral poor lighting fails. Although only a small number of cameras and lighting conditions are used in this experiment, the results reflect the findings from the simulated experiments performed on a much larger set of sensors and light sources.

We also perform correction at 500 lux for D65, incandescent, and fluorescent lighting

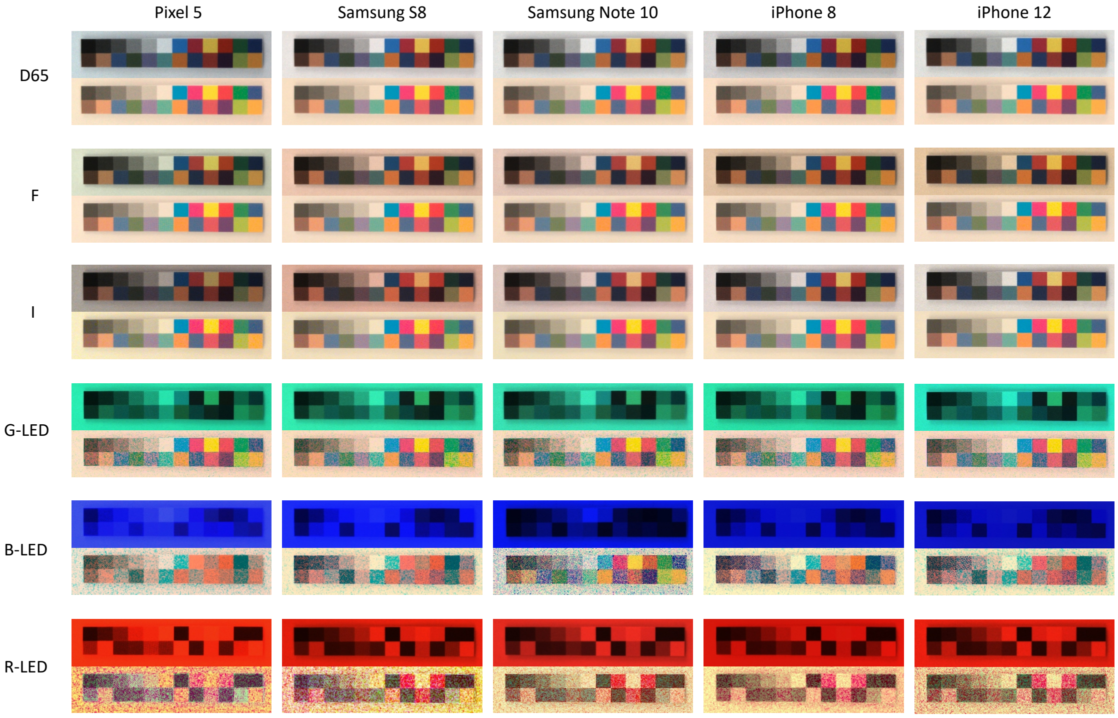


Figure 4.3: The custom calibration chart captured by 5 different cameras under the lights: D65, incandescent (I), fluorescent (F), and the tunable LED set to green (G), red (R), and blue (B). The images are in pairs where the top image is the raw image scaled between 0-255 for display and the bottom image is the corrected image.

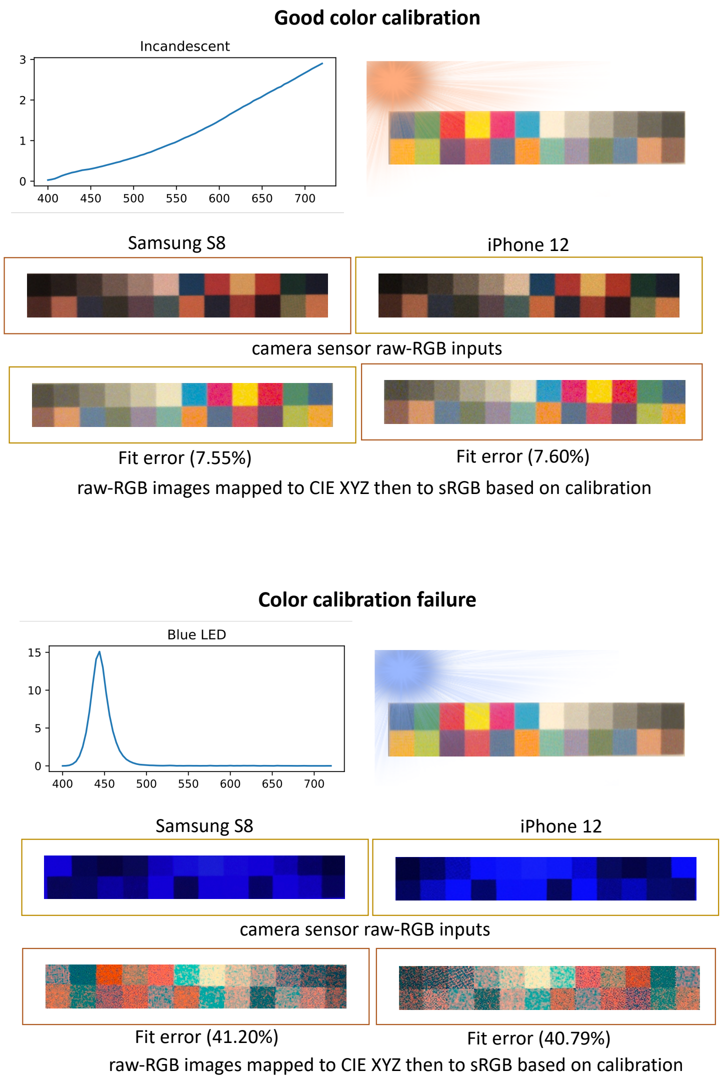


Figure 4.4: This figure shows examples of real camera captured images during lighting that results in “good” and “failed” color calibration. In a home-care application, the spectral sensitivity of the sensor and lighting would not be known. From our analysis, failure for color calibration is attributed to the lighting being spectral poor vs. a degenerated combination of sensor and light source.

Table 4.2: Color calibration errors (in terms of % fit error) for different smartphones and lights at 100 lux using the custom calibration chart. The first column lists the phones used, the other five columns are calibration errors (in percentage) for the lights: D65, incandescent (I), fluorescent (F), and the tunable LED set to green (G), red (R), and blue (B).

Phone	D65	I	F	G-LED	R-LED	B-LED
Pixel 5	5.08	6.10	7.87	8.34	33.54	38.36
Samsung S8	5.26	6.19	7.45	9.14	30.16	41.20
Samsung Note 10	7.73	8.04	6.75	12.24	28.77	16.09
Apple iPhone 8	5.65	5.35	5.95	9.36	35.05	30.07
Apple iPhone 12	7.48	6.71	6.83	9.84	36.93	40.79

Table 4.3: Color calibration errors (in terms of Delta E 2000) for different smartphones and lights at 100 lux using the custom calibration chart. The first column lists the phones used, the other five columns are calibration errors for the lights: D65, incandescent (I), fluorescent (F), and the tunable LED set to green (G), red (R), and blue (B).

Phone	D65	I	F	G-LED	R-LED	B-LED
Pixel 5	2.10	2.27	1.92	4.05	20.57	18.13
Samsung S8	1.92	2.85	2.11	5.73	15.05	19.58
Samsung Note 10	2.18	2.27	2.39	7.25	15.69	10.36
Apple iPhone 8	2.14	2.09	2.14	6.45	18.47	16.94
Apple iPhone 12	2.42	2.59	2.26	5.52	17.33	18.72

conditions to show the correction in more normal lighting conditions. The results for this analysis are in Table 4.4 and 4.5. All lighting condition and sensor combinations resulted in good calibration for both percent and Delta E error metrics.

We also display the raw and corrected image pairs captured with the custom chart in Fig. 4.3. This figure shows a visual representation of how we can correct various different lighting conditions and camera images to a common space. Although the top raw image varies quite a bit between setups, the bottom corrected image is visually indistinguishable

Table 4.4: Color calibration errors (in terms of % fit error) for different smartphones and lights at 100 lux using the custom calibration chart. The first column lists the phones used, the other three columns are calibration errors (in percentage) for the lights: D65, incandescent (I), and fluorescent (F).

Phone	D65	I	F
Pixel 5	5.49	7.54	6.43
Samsung S8	5.85	6.18	6.43
Samsung Note 10	7.10	8.33	7.42
Apple iPhone 8	5.58	5.35	6.04
Apple iPhone 12	7.14	7.54	7.71

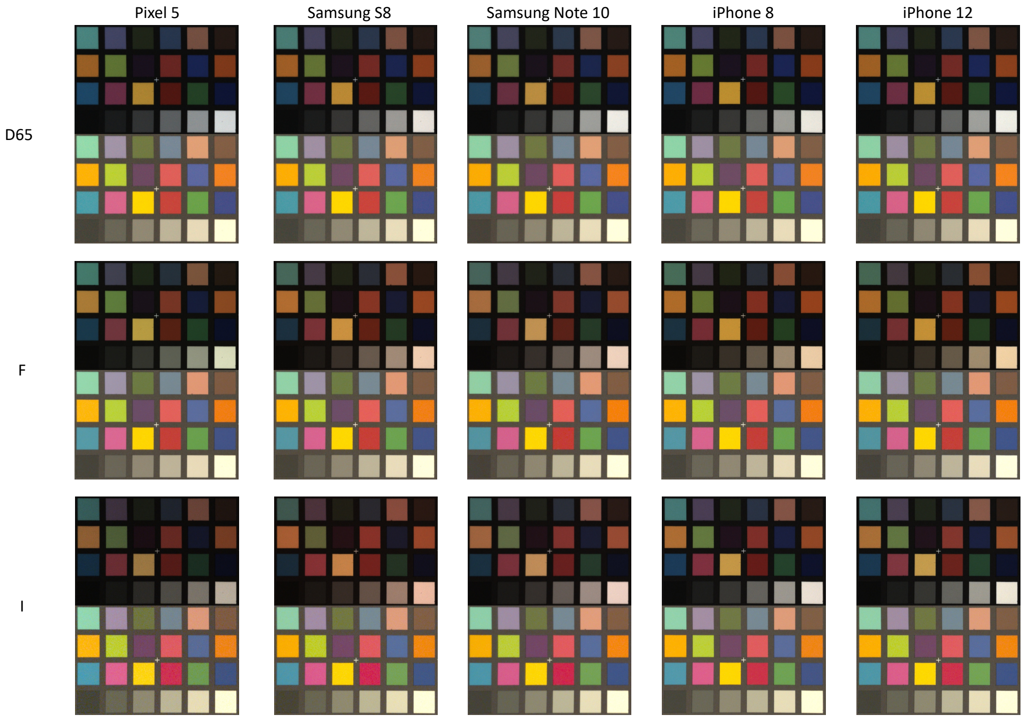
Table 4.5: Color calibration errors (in terms of Delta E 2000) for different smartphones and lights at 500 lux using the custom calibration chart. The first column lists the phones used, the other three columns are calibration errors for the lights: D65, incandescent (I),and fluorescent (F).

Phone	D65	I	F
Pixel 5	2.08	2.66	1.92
Samsung S8	1.85	2.67	2.11
Samsung Note 10	2.23	2.67	2.56
Apple iPhone 8	2.09	1.79	1.91
Apple iPhone 12	2.19	2.34	2.47

in all normal lighting.

4.2.2 X-Rite Calibration Chart

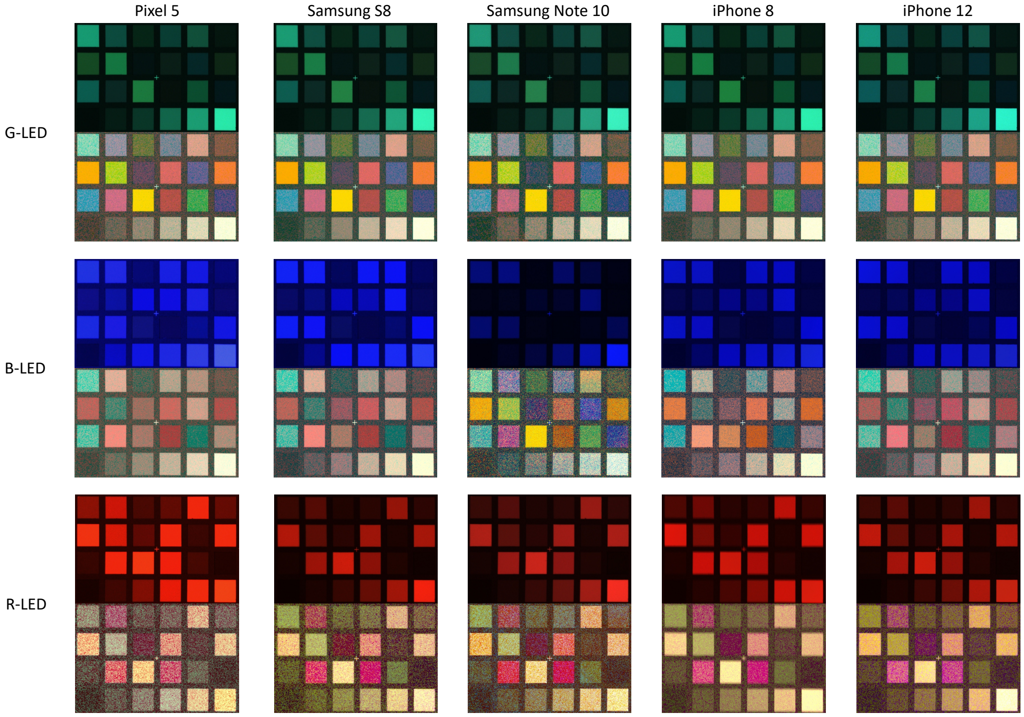
We also show Tables 4.6 - 4.9 displaying calibration errors when using an X-Rite Passport color chart. As expected, the X-Rite Passport performs well and calibration is good under standard lighting. The X-Rite color checker performs slightly better in terms of color calibration error on most of the cameras. We hypothesize this is due to the fact that the



(a) Lights: D65, incandescent (I), and fluorescent (F)

Figure 4.5: The X-Rite calibration chart captured by 5 different cameras under the lights: D65, incandescent (I), and fluorescent (F). The images are in pairs where the top image is the raw image scaled between 0-255 for display and the bottom image is the corrected image.

pigments are more spectrally diverse then the colors from the custom calibration pattern. The errors also might simply just be due to the set of colors chosen. Since, the calibration of each camera is done with a rigid linear transformation, the errors are highly dependent on the colors used to find the transform. The important takeaway is that both the X-Rite and custom chart achieve errors that are good for all standard lighting conditions. These results show that a custom calibration chart created with a standard office printer can be



(b) Lights: green (G), red (R), and blue (B))

Figure 4.5: The X-Rite calibration chart captured by 5 different cameras under the lights: tunable LED set to green (G), red (R), and blue (B). The images are in pairs where the top image is the raw image scaled between 0-255 for display and the bottom image is the corrected image.

used as an effective stand-in for an X-Rite color chart.

We also display the raw and corrected image pairs captured with the X-Rite chart in Fig. 4.5. Another interesting observation is that for images taken under the green LED, a correction was performed reasonably well. Although, the colorimetric mapping was not as accurate as other light sources, the results were typically not considered failures. In fact, after colorimetric mapping, the color chart illuminated by green LED lights appeared

Table 4.6: Color calibration errors (in terms of % fit error) for different smartphones and lights at 100 lux using the X-Rite calibration chart. The first column lists the phones used, the other five columns are calibration errors (in percentage) for the lights: D65, incandescent (I), fluorescent (F), and the tunable LED set to green (G), red (R), and blue (B).

Phone	D65	I	F	G-LED	R-LED	B-LED
Pixel 5	6.02	6.39	7.80	9.57	42.49	37.04
Samsung S8	5.01	4.92	5.46	10.37	41.14	37.03
Samsung Note 10	5.12	7.06	6.21	12.73	39.00	15.96
Apple iPhone 8	4.66	4.45	5.30	9.21	41.20	31.74
Apple iPhone 12	6.27	5.32	7.37	9.36	42.08	35.70

Table 4.7: Color calibration errors (in terms of % delta E 2000) for different smartphones and lights at 100 lux using the X-Rite calibration chart. The first column lists the phones used, the other five columns are calibration errors for the lights: D65, incandescent (I), fluorescent (F), and the tunable LED set to green (G), red (R), and blue (B).

Phone	D65	I	F	G-LED	R-LED	B-LED
Pixel 5	2.36	2.97	2.73	3.64	14.02	15.53
Samsung S8	2.44	3.75	3.00	3.80	12.22	15.63
Samsung Note 10	2.09	3.60	2.72	5.44	12.33	9.39
Apple iPhone 8	2.41	2.82	3.35	4.03	13.67	15.43
Apple iPhone 12	2.10	3.37	2.72	3.92	13.11	15.49

Table 4.8: Color calibration errors (in terms of % fit error) for different smartphones and lights at 500 lux using the X-Rite calibration chart. The first column lists the phones used, the other three columns are calibration errors (in percentage) for the lights: D65, incandescent (I), and fluorescent (F).

Phone	D65	I	F
Pixel 5	5.47	6.43	6.20
Samsung S8	6.10	5.60	5.46
Samsung Note 10	5.12	4.93	6.21
Apple iPhone 8	4.58	3.90	4.90
Apple iPhone 12	6.04	5.07	5.88

Table 4.9: Color calibration errors (in terms of Delta E 2000) for different smartphones and lights at 500 lux using the X-Rite calibration chart. The first column lists the phones used, the other three columns are calibration errors for the lights: D65, incandescent (I), and fluorescent (F)

Phone	D65	I	F
Pixel 5	2.48	2.82	2.57
Samsung S8	2.30	3.20	2.57
Samsung Note 10	2.18	2.74	2.49
Apple iPhone 8	2.33	2.82	2.60
Apple iPhone 12	2.10	2.84	2.27

visually correct. When examining the green LED spectral profile, the light source contains information throughout all the visual spectrum but tapers out in intensity towards the blue and red wavelengths. Since the light covers all of the visible spectrum throughout most wavelengths, it was possible to produce adequate correction can be done that provides us with information about the scene. Almost all real lighting conditions have some signal throughout all wavelengths in the visual spectrum, thus the green LED correction provides us an example of how correction looks like with weaker signal in parts of the visual spectrum.

The red LED and blue LED lighting conditions are not possible to correct because there is missing signal in many wavelengths. We notice there are many illumination caused metamers in the red and blue lighting images. Patches with very different reflectances look the same because the illumination is missing a large portion of the visible wavelengths.

4.3 Summary

Our findings from both synthetic and real images indicated that color imaging failure is rare. We did not observe specific sensor-illumination pairs that resulted in color failure. Instead, When color imaging failures did occur, it is attributed to poor spectral lighting and impacted most of the camera sensors.

Chapter 5

Conclusion

This thesis has analyzed color calibration failure for consumer cameras. Our work assumes a well-lit environment, proper focus and exposure, and low image noise. Within this scenario we sought to answer two questions: (1) How likely is failure to occur and (2) If color calibration does fail, what was the cause? To this end, we examined 97 camera sensors and 108 light sources and found that no combination of sensor and light source resulted in color calibration failure for 105 of the lights. The only failure cases were attributed to three light conditions generated by a tunable commercial colors. These tuned lights resulted in consistent failure across *all* sensors due to the spectral quality of the light source (mainly narrow band). Additional experiments were performed on a sample of real smartphone and commercial lights using a custom low-cost calibration pattern and an X-Rite chart with a reasonable diversity in CIE XYZ values. While it is well-known that spectral poor illumination results in poor color reproduction, our analysis shows that it is rare to encounter specific sensor and lighting combinations that lead to erroneous color imaging on consumer cameras. Our analysis indicates that color failure is rare for most sensor and lighting combinations and when failure does occur, the light source is the most likely culprit. The implication of our finding is that developers of apps that rely on color can assume that most sensor and lighting combinations will result in good color calibration. When color calibration failure does occur, the application can direct users to seek a better lighting environment.

A clear area for future work is to examine more smartphone cameras and commercial lighting sources. In addition, image capture in a home environment is susceptible to problems such as shadows and non-uniform illumination. Our approach also assumed the color chart was illuminated by a single dominant light source. Color imaging under mixed illumination warrants further investigation.

Bibliography

[1] S. D. Kim, Y. Koo, and Y. Yun, "A smartphone-based automatic measurement method for colorimetric ph detection using a color adaptation algorithm," *Sensors*, vol. 17, no. 7, 2017.

[2] Kap, V. Kılıç, J. G. Hardy, and N. Horzum, "Smartphone-based colorimetric detection systems for glucose monitoring in the diagnosis and management of diabetes," *Analyst*, vol. 146, pp. 2784–2806, 2021.

[3] E. J. Wang, J. Zhu, M. Jain, T.-J. Lee, E. Saba, L. Nachman, and S. N. Patel, "Seismo: Blood pressure monitoring using built-in smartphone accelerometer and camera," in *ACM CHI - Conference on Human Factors in Computing Systems*, p. 1–9, 2018.

[4] K. Xiao, Y. Zhu, C. Li, D. Connah, J. M. Yates, and S. Wuerger, "Improved method for skin reflectance reconstruction from camera images," *Optics Express*, vol. 24, pp. 14934–14950, Jun 2016.

[5] H. Otaka, H. Shimakura, and I. Motoyoshi, "Perception of human skin conditions and image statistics," *Journal of Optical Society of America - A*, vol. 36, pp. 1609–1616, Sep 2019.

[6] D. Akkaynak, T. Treibitz, B. Xiao, U. Gürkan, J. Allen, U. Demirci, and R. Hanlon, "Use of commercial off-the-shelf digital cameras for scientific data acquisition and scene-specific color calibration," *Journal of the Optical Society of America. A*, vol. 31, pp. 312–21, 02 2014.

[7] H. C. Karaimer and M. S. Brown, "A software platform for manipulating the camera imaging pipeline," in *European Conference on Computer Vision*, pp. 429–444, 2016.

[8] A. Chakrabarti, D. Scharstein, and T. Zickler, "An empirical camera model for internet color vision," in *British Machine Vision Conference*, 2009.

[9] G. D. Finlayson, M. Mohammadzadeh Darrodi, and M. Mackiewicz, "The alternating least squares technique for nonuniform intensity color correction," *Color Research & Application*, vol. 40, no. 3, pp. 232–242, 2015.

[10] G. Hong, M. R. Luo, and P. A. Rhodes, "A study of digital camera colorimetric characterization based on polynomial modeling," *Color Research & Application*, vol. 26, no. 1, pp. 76–84, 2001.

[11] S. Tedla, Y. Wang, M. Patel, and M. S. Brown, "Analyzing color imaging failure on consumer-grade cameras," *Journal of Optical Society of America - A*, vol. 39, no. 6, pp. B21–B27, 2022.

[12] D. Sliney, "What is light? the visible spectrum and beyond," *Eye*, vol. 30, no. 2, pp. 222–229, 2016.

[13] P. Ronan, "Revised diagram with re-aligned spectrum - file:empectrumrevised.png," Feb 2013.

[14] "Blue Flower." `https://freesvg.org/odysseus-blue-flower`.

[15] "Wool Image." `https://freesvg.org/wool`.

[16] "Vector Image of Candy." `https://www.publicdomainvectors.org/en/free-clipart/Vector-image-of-candy-in-a-wrapper-picture/31861.html`.

[17] D. A. Kerr, "The CIE XYZ and xyY color spaces," *Colorimetry*, vol. 1, no. 1, pp. 1–16, 2010.

[18] J. Schnapf, T. Kraft, and D. Baylor, "Spectral sensitivity of human cone photoreceptors," *Nature*, vol. 325, no. 6103, pp. 439–441, 1987.

[19] A. Stockman, D. I. MacLeod, and N. E. Johnson, "Spectral sensitivities of the human cones," *Journal of Optical Society of America - A*, vol. 10, no. 12, pp. 2491–2521, 1993.

[20] H. C. Karaimer and M. S. Brown, "Improving color reproduction accuracy on cameras," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] K. Barnard, L. Martin, B. Funt, and A. Coath, "A data set for color research," *Color Research and Application*, vol. 27, pp. 147–151, 2002.

[22] D. A. Rowlands, "Color conversion matrices in digital cameras: A tutorial," *Optical Engineering*, vol. 59, no. 11, p. 110801, 2020.

[23] P.-C. Hung, "Colorimetric calibration in electronic imaging devices using a look-up-table model and interpolations," *Journal of Electronic Imaging*, vol. 2, no. 1, pp. 53–61, 1993.

[24] D. H. Brainard, P. Longère, P. B. Delahunt, W. T. Freeman, J. M. Kraft, and B. Xiao, "Bayesian model of human color constancy," *Journal of Vision*, vol. 6, no. 11, pp. 10–10, 2006.

[25] M. Brown, "Understanding color and the in-camera image processing pipeline for computer vision," *ICCV 2019 tutorial: Understanding color and the in-camera image processing pipeline for Computer Vision*, Oct 2019.

[26] "Female Head-Younger PNG icon." `iconspng.com/image/38060/female-headyounger`.

[27] H. C. Karaimer and M. S. Brown, "A software platform for manipulating the camera imaging pipeline," in *European Conference on Computer Vision*, 2016.

[28] "Glacier National Park Image." `https://www.flickr.com/photos/usinterior/49942081792`.

[29] J. I. S. Silva, G. G. Carvalho, M. S. Santos, D. J. Santiago, L. P. de Albuquerque, J. F. P. Battle, G. M. da Costa, and T. I. Ren, "A deep learning approach to mobile camera image signal processing," in *Anais Estendidos do XXXIII Conference on Graphics, Patterns and Images*, pp. 225–231, SBC, 2020.

[30] V. D. Silva, V. Chesnokov, and D. Larkin, "A novel adaptive shading correction algorithm for camera systems," *Electronic Imaging*, vol. 2016, no. 18, pp. 1–5, 2016.

[31] J. Wang, F. Shi, J. Zhang, and Y. Liu, "A new calibration model of camera lens distortion," *Pattern Recognition*, vol. 41, no. 2, pp. 607–615, 2008.

[32] B. Atcheson, F. Heide, and W. Heidrich, "CALtag: High precision fiducial markers for camera calibration.," in *Vision, Modeling, and Visualization*, vol. 10, pp. 41–48, 2010.

[33] S.-H. Lam and C.-W. Kok, "Demosaic: Color filter array interpolation for digital cameras," in *Pacific-Rim Conference on Multimedia*, pp. 1084–1089, Springer, 2001.

[34] Y. Zhao, L.-M. Po, T. Zhang, Z. Liao, X. Shi, Y. Zhang, W. Ou, P. Xian, J. Xiong, C. Zhou, and W. Y. Yu, "Saliency map-aided generative adversarial network for raw to rgb mapping," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3449–3457, 2019.

[35] B. E. Bayer, "Color imaging array," *United States Patent 3,971,065*, 1976.

[36] K. T. Mullen, "The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings.," *The Journal of Physiology*, vol. 359, no. 1, pp. 381–400, 1985.

[37] H. Ma, S. Liu, G. Wei, Z. Tang, S. Zhang, and Y. Liu, "A new color interpolation algorithm for bayer pattern digital cameras based on green components," *Lecture Notes in Electrical Engineering*, vol. 238, pp. 1103–1108, 06 2014.

[38] J. E. Adams Jr, "Interactions between color plane interpolation and other image processing functions in electronic photography," in *Cameras and Systems for Electronic Photography and Scientific Imaging*, vol. 2416, pp. 144–151, SPIE, 1995.

[39] R. Lukac and K. Plataniotis, “A normalized model for color-ratio based demosaicking schemes,” *International Conference on Image Processing*, vol. 3, pp. 1657 – 1660, 11 2004.

[40] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander, “Demosaicking methods for bayer color arrays,” *Journal of Electronic Imaging*, vol. 11, pp. 306–315, 2002.

[41] R. A. M. Jr., S. S. Young, J. Reynolds, K. Krapels, J. Fanning, and T. Corbin, “Review of Bayer pattern CFA demosaicing with new quality assessment algorithms,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXI* (G. C. Holst and K. A. Krapels, eds.), vol. 7662, pp. 363 – 374, International Society for Optics and Photonics, SPIE, 2010.

[42] J. E. Adams Jr and J. F. Hamilton Jr, “Adaptive color plane interpolation in single sensor color electronic camera,” *United States Patent 5,652,621*, July 29 1997.

[43] H.-A. Chang and H. Chen, “Directionally weighted color interpolation for digital cameras,” in *2005 IEEE International Symposium on Circuits and Systems*, pp. 6284–6287, IEEE, 2005.

[44] T.-y. Jung, S. Yang, J. Lee, and J. Jeong, “Enhanced directionally weighted demosaicing for digital cameras,” in *2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, pp. 518–522, 2008.

[45] R. K. Thakur, A. Tripathy, and A. K. Ray, “A new edge adaptive and directionally weighted color interpolation algorithm of digital camera images,” in *Proceedings of*

the International Conference on Advances in Computing, Communication and Control, pp. 397–402, 2009.

[46] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 132–141, 2007.

[47] M. Afifi, A. Abdelhamed, A. Abuolaim, A. Punnappurath, and M. S. Brown, "CIE XYZ net: Unprocessing images for low-level computer vision tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[48] H. Can and M. Brown, "Beyond raw-rgb and srgb: Advocating access to a colorimetric image state," *Color and Imaging Conference*, vol. 2019, pp. 86–90, 10 2019.

[49] J. T. Barron and Y.-T. Tsai, "Fast fourier color constancy," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 886–894, 2017.

[50] M. E. Solmaz, A. Y. Mutlu, G. Alankus, V. Kılıç, A. Bayram, and N. Horzum, "Quantifying colorimetric tests using a smartphone app based on machine learning classifiers," *Sensors and Actuators B: Chemical*, vol. 255, pp. 1967–1973, 2018.

[51] "Winter Forest Image." `https://www.publicdomainpictures.net/pictures/20000/velka/winter-forest.jpg`.

[52] D. Cheng, B. Price, S. Cohen, and M. S. Brown, "Beyond white: Ground truth colors for color constancy correction," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 298–306, 2015.

[53] C.-C. Weng, H. Chen, and C.-S. Fuh, “A novel automatic white balance method for digital still cameras,” in *2005 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3801–3804 Vol. 4, 2005.

[54] M. Afifi, B. Price, S. Cohen, and M. S. Brown, “When color constancy goes wrong: Correcting improperly white-balanced images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1535–1544, 2019.

[55] E. Lam, “Combining gray world and retinex theory for automatic white balance in digital photography,” in *Proceedings of the Ninth International Symposium on Consumer Electronics, 2005. (ISCE 2005).*, pp. 134–139, 2005.

[56] G. Zapryanov and I. Nikolova, “Automatic white balance algorithms for digital still cameras - a comparative study,” *Information Technologies and Control*, vol. 1, pp. 16–22, 01 2012.

[57] R. Ramanath, W. Snyder, Y. Yoo, and M. Drew, “Color image processing pipeline,” *Signal Processing Magazine, IEEE*, vol. 22, pp. 34 – 43, 02 2005.

[58] Y. Salih, W. bt. Md-Esa, A. S. Malik, and N. Saad, “Tone mapping of HDR images: A review,” in *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)*, vol. 1, pp. 368–373, IEEE, 2012.

[59] C. A. Poynton, “SMPTE tutorial:“gamma” and its disguises: The nonlinear mappings of intensity in perception, CRTS, film, and video,” *SMPTE Motion Imaging*, vol. 102, no. 12, pp. 1099–1108, 1993.

[60] G. Starkweather, “Colorspace interchange using sRGB,” *White Paper available at http://www. srgb. com*, p. 19, 1998.

[61] C. A. Poynton, “Rehabilitation of gamma,” in *Human Vision and Electronic Imaging III*, vol. 3299, pp. 232–249, International Society for Optics and Photonics, 1998.

[62] G. M. Johnson and M. D. Fairchild, “Wool description of S-CIELAB and CIEDE2000,” *Color Research & Application*, vol. 28, no. 6, pp. 425–435, 2003.

[63] J. Jiang, D. Liu, J. Gu, and S. Süsstrunk, “What is the space of spectral sensitivity functions for digital color cameras?,” in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 168–179, 2013.

[64] ImageEngineering. `https://www.image-engineering.de/` [accessed: Oct 18, 2021].

[65] Chromaxion. `https://chromaxion.com/spectral-library.php` [accessed: Jan 22, 2021].

[66] AdobeSystemsIncorporated, “Digital negative specification version 1.4.0.0,” Jun 2012.

[67] A. Abdelhamed, S. Lin, and M. S. Brown, “A high-quality denoising dataset for smartphone cameras,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, 2018.

[68] A. Kordecki and H. Palus, “Automatic detection of colour charts in images,” *Przeglad Elektrotech*, vol. 90, pp. 197–202, 2014.

[69] I. Lebedev, A. Erashov, and A. Shabanova, “Accurate autonomous UAV landing using vision-based detection of aruco-marker,” in *International Conference on Interactive Collaborative Robotics*, pp. 179–188, Springer, 2020.