A MATLAB Toolbox for Color Space Conversion, Calibration, Visualization, and Color Vision Deficiency Simulation

by

Zheng Zhang

A Master Report Submitted to the Faculty of the

JAMES C. WYANT COLLEGE OF OPTICAL SCIENCES

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2023

TABLE OF CONTENTS

ABSTRACT

This thesis aims to develop a practical MATLAB toolbox designed to facilitate color space conversion, color calibration, color space visualization, and color vision deficiency simulation. The toolbox will provide a user-friendly, versatile, and powerful tool to manipulate and analyze color information effectively, ultimately enhancing the study of color vision, color processing, and the development of accessible color schemes for various applications.

1. INTRODUCTION

1.1. Background and motivation

1.1.1. Importance of color spaces in digital imaging, computer vision, and design

a) **Representation and consistency**: Different color spaces offer alternative ways of representing and encoding color information. They ensure that color data can be consistently represented and reproduced across different devices and platforms. For example, the RGB color space is widely used in digital displays, while the CMYK color space is used for printing purposes.

b) **Perceptual uniformity**: Some color spaces, such as CIE-Lab and CIE-Luv, are designed to be more perceptually uniform, meaning that the distances between colors in these

spaces closely align with human perception of color differences. This is important for applications like image editing and color palettes, where maintaining the perceptual relationships between colors is essential.

c) **Processing efficiency**: Color spaces can be optimized for specific tasks or applications, resulting in more efficient processing. For example, the HSV and HSL color spaces separate color information (hue) from intensity (value or lightness), simplifying tasks like adjusting brightness or contrast.

d) **Specialized applications**: Some color spaces are designed for specific applications, such as YCbCr for video compression and YIQ for analog television broadcasting. These color spaces exploit certain properties of human vision, like sensitivity to different color channels, to optimize performance and quality for their specific use cases.

1.1.2. Importance of color calibration

a) **Consistency and accuracy**: Color calibration is essential for ensuring that colors are accurately and consistently represented across different devices, such as cameras, monitors, and printers. Without calibration, the same color may appear differently on different devices, leading to inconsistencies and inaccuracies in color reproduction.

b) **Professional applications**: In industries like graphic design, photography, and printing, color accuracy is critical. Color calibration helps professionals ensure that the

colors in their work are accurately represented, both on-screen and in print, maintaining the integrity of their designs and images.

c) **User experience**: Accurate color reproduction is important for the overall user experience, as it allows users to view and edit images, videos, and designs with confidence, knowing that the colors they see on their devices will closely match the final output.

d) **Color management**: Color calibration is a key component of color management workflows, which aim to maintain consistent color representation across different devices, software, and output formats. By calibrating devices and using color profiles, users can achieve a higher level of control over color reproduction, ensuring that their work is consistently and accurately represented.

1.2. Objective and scope of the study

The primary objective of this study is to design, implement, and validate a MATLAB toolbox that helps research in the lab. Initially, the toolbox only contains functions of color space conversion and color calibration to simplify the repeated labor work. However, with further study in the OPTI 588 course, I was inspired by projects undertaken by my peer, Ryan Hamilton, who made a neat color space visualizer. Consequently, I decided to enhance my toolbox by adding features of color space visualization and color vision deficiency simulation.

1.3. Overview of the thesis

This thesis embarks on a journey to develop a MATLAB toolbox that streamlines color space conversion and color difference calculator, offers a simple way of color calibration for display, and provides intuitive visualization of color gamut and diagram. Furthermore, this toolbox aims to simulate dichromatic color perception, as the fundamental move to re-color the image to correct the color-blind vision.

The structure of this thesis is organized as follows; Section 2 gives review of common color space models, color calibration and color vision deficiencies. Section 3 introduces the principles and algorithms of the toolbox. Section 4 shows the result of the toolbox and its evaluation. Then the conclusion is given in Section 5.

2. REVIEW of RELEVANT LITERATURE

2.1. Color space models

A color space is a specific organization or representation of colors, allowing for consistent communication of color information across varied platforms, devices, and applications. It essentially serves as a framework within which colors can be quantified, related, and interpreted.

There are multiple color space models to adapt different situation. The inception of color spaces can be traced back to the groundbreaking work of the Commission Internationale de

l'éclairage (CIE), which introduced the CIE XYZ color space in 1931. This mathematical model became a bedrock for many subsequent color spaces. The CIE xyY model which explains chromaticity and luminance directly and CIE Lab and Luv models which give perceptually uniform color spaces are also developed.

The color space can also be device dependent. For digital media, the RGB (Red, Green, Blue) model is paramount, predicated on an additive color mixing principle suited for electronic displays like monitors and TVs. Conversely, the CMYK (Cyan, Magenta, Yellow, Black) space is foundational in color printing, operating on a subtractive color mixing approach. Different color mixing principles decide their different applications.

HSV and HSL color space describe the cylindrical-coordinate representations of points in an RGB color model, with the component becoming Hue, Saturation and Value/Lightness. There are also some color spaces designed for specific applications, like YCbCr color space for video compression.

2.2. Color calibration techniques for display

There are several color calibration methods for display, including the use of lookup tables (LUTs), matrix-based methods, and gamma correction. If some assumptions are made, a color calibration can be done easily by measure the tone response curves (TRCs) of the display. While the LUTs method requires lots of data capacity and computation time and gamma correction is lack of freedom, the TRC based method allows for the characterization of the

display's response to different input levels, which can be used to predict the display's output for any given input.

2.3. Color vision deficiencies and simulation

Color vision deficiencies (CVD), commonly referred to as color blindness, encompass a range of disorders that affect an individual's ability to perceive colors accurately. While many perceive the world through a trichromatic lens, with receptors for red, green, and blue light, those with CVD may experience deficiencies in one or more of these receptors.

2.3.1. Types of color vision deficiencies

The typical human eye houses three distinct types of cone cells—L, M, and S—each attuned to varying segments of the visible spectrum. Broadly, color vision deficiencies (CVD) manifest in three primary forms:

a) **Monochromacy**: This is an exceedingly rare condition. Characterized by the functionality of just one cone type, individuals with this deficiency are typically completely color-blind, though they might still possess one perceptual pathway in addition to the rod pathway.

b) **Dichromacy**: In this condition, individuals lack one of the cone photopigments, resulting in perception through only two channels. Depending on which type of cone is missing, dichromacy can also be classified as Protanomaly and Protanopia (Red

weaknesses), Deuteranomaly and Deuteranopia (Green weaknesses) and Tritanomaly and Tritanopia (Blue weaknesses).

c) **Anomalous Trichromacy**: Here, while all three cone photopigments are present, one of them is anomalous—with its peak sensitivity shifted.

2.3.2. Existing simulation methods

Brettel proposed an algorithm that transforms a digitized color image to simulate the appearance of the image for people who have dichromatic forms of color blindness. The algorithm replaces each stimulus by its projection onto a reduced stimulus surface, which is defined by a neutral axis and by the LMS locations of those monochromatic stimuli that are perceived as the same hue by normal trichromats, and a given type of dichromat. The dichromat's color confusions are deduced from colorimetry, and the residual hues in the transformed image are derived from the reports of unilateral dichromats described in the literature. The simulation algorithm is expressed in terms of transformations of the three-dimensional LMS space. Furthermore, J. Ruminski also gives a detailed transformation involving LMS space to simulate dichromats vision.

3. TOOLBOX DESIGN and IMPLEMENTATION

3.1. Modular architecture

The toolbox features a modular design, as illustrated in Figure 1. Although certain functions facilitate interactions between modules, it's important to note that each module must

be accessed through the main menu interface, depicted in Figure 2. There are four main modules: Color Space Conversion, Color Calibration, Color Space Visualizer, and CVD Simulation. Also, a pop-up display window is designed for fluent measurements.

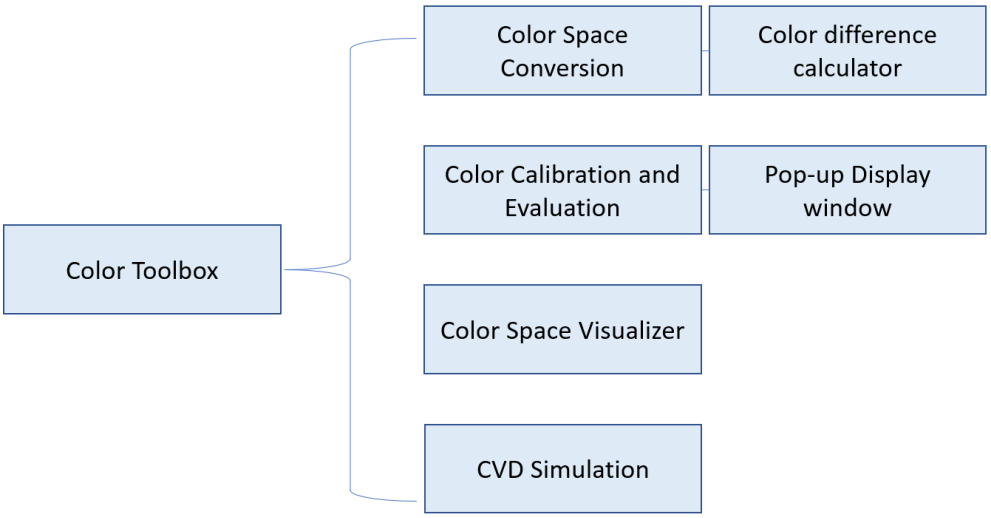

Figure 1. Overall structure of the color toolbox

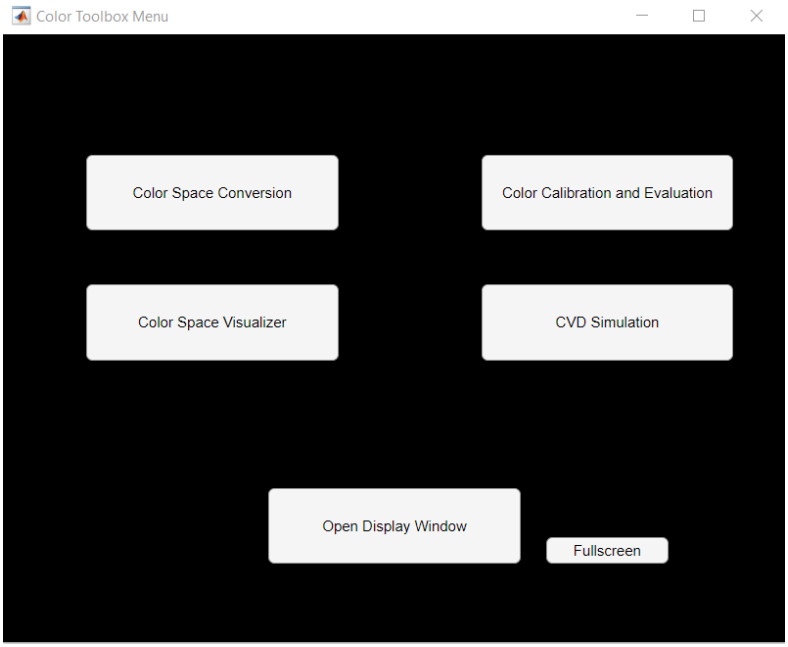

Figure 2. Color toolbox menu layout

3.1.1. Color space conversion module

Figure 3. Color space conversion module layout

This module will detect the latest input coordinates and transfer them into other color spaces based on conversion settings, like illuminant and RGB bit depth, and display them at the same time. Furthermore, the color temperature will be calculated. Beyond that, color difference in LAB space can be calculated by saving two colors as color 1 and color 2 from the color space convertor.

3.1.2. Color calibration module

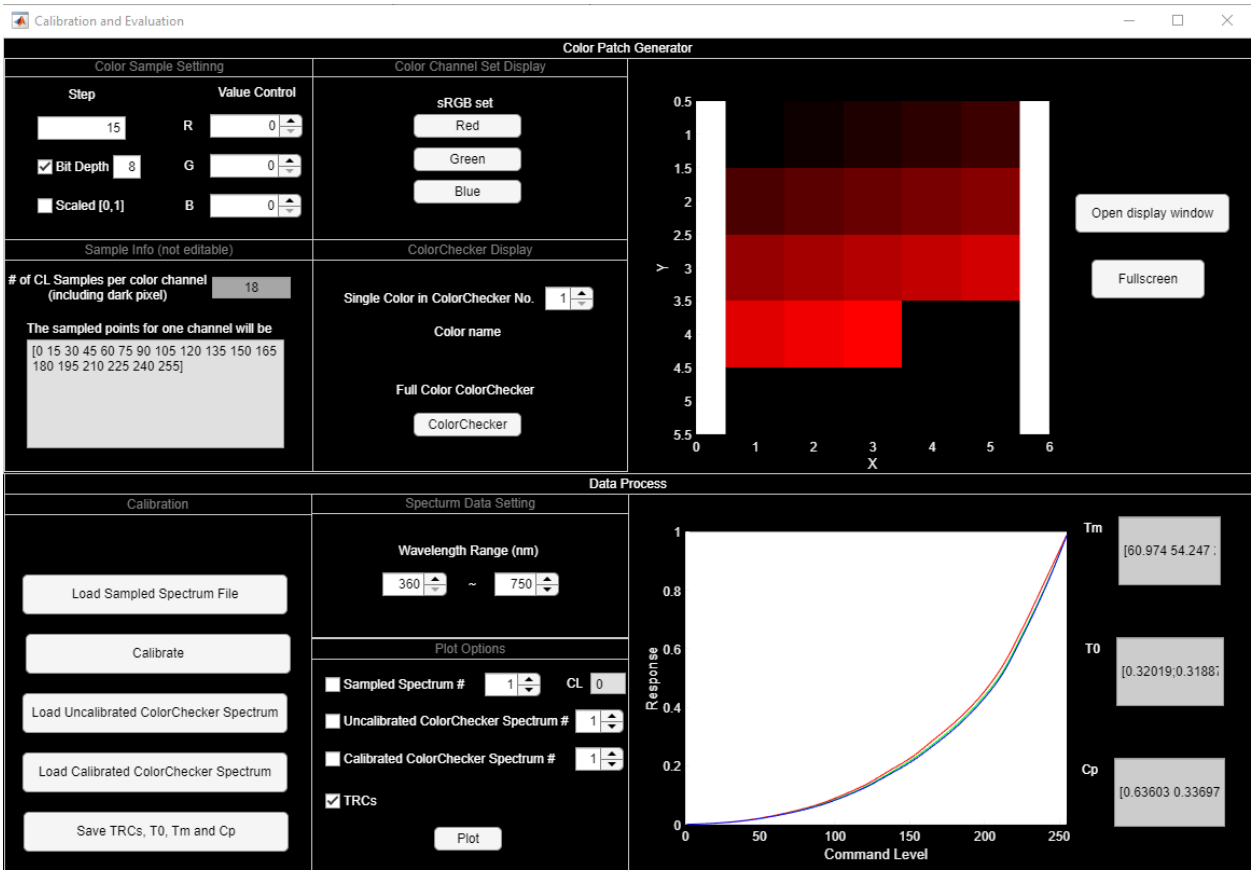

Figure 4. Color calibration module layout

This module is designed for basic color calibration of monitors. It can provide several image sets, like one-channel samples or color checker, to be displayed on the monitor for measurement, and then calculate and save the tone response curves and calibration matrix.

3.1.3. Color space visualizer module

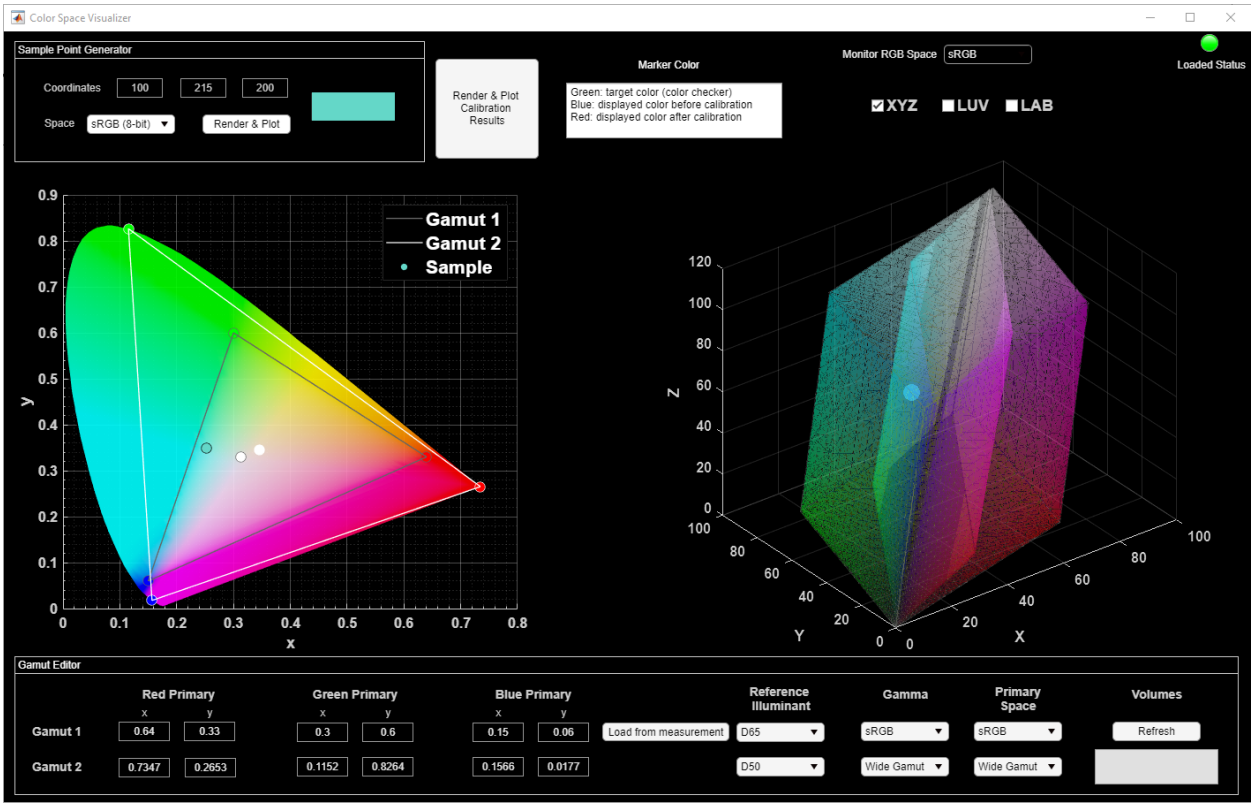

Figure 5. Color space visualizer module layout

This module will display a 2D x-y color diagram and 3D XYZ, LUV or LAB color volume of user-defined color gamut. Also, one sample point can be put inside the plots. Furthermore, the result from calibration module can be displayed here directly.

3.1.4. Color vision deficiency simulation module

This module will simulate the three dichromats vision (protanopia, deuteranopia and tritanopia) for the loaded image.

3.2. Algorithm and implementation

3.2.1. Color space conversion algorithms

XYZ color space is chosen as the intermediary color space for the overall conversion, any non-XYZ color coordinates input will be transferred to the XYZ color space first and then calculated to all other color spaces. Some actions are also required during the conversion, including chromatic adaptation (or white adaptation), gamma correction and bit depth scaling for RGB and Y scaling for XYZ.

a) Chromatic adaptation

This adaptation is necessary when the input illuminant and output illuminant does not match. Here the Bradford method is used. The XYZ coordinates of the input reference illuminant are denoted as (X_{r0}, Y_{r0}, Z_{r0}) and the output reference illuminant XYZ coordinates are denoted as (X_r, Y_r, Z_r). The input XYZ coordinates are (X_0, Y_0, Z_0) and the output XYZ coordinates are (X, Y, Z). The equation is

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathrm{M}_{BM} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix},$$

Where M_{BM} is a known matrix from the two illuminant XYZ coordinates,

$$\mathrm{M}_{BM} = \begin{bmatrix} 0.8951 & 0.2664 & -0.1614 \\ -0.7502 & 1.7135 & 0.0367 \\ 0.0389 & -0.0685 & 1.0296 \end{bmatrix}^{-1} \begin{bmatrix} \frac{X_r}{X_{r0}} & 0 & 0 \\ 0 & \frac{Y_r}{Y_{r0}} & 0 \\ 0 & 0 & \frac{Z_r}{Z_{r0}} \end{bmatrix} \begin{bmatrix} 0.8951 & 0.2664 & -0.1614 \\ -0.7502 & 1.7135 & 0.0367 \\ 0.0389 & -0.0685 & 1.0296 \end{bmatrix}$$

b) XYZ to other unrendered color spaces

The XYZ coordinates should have the Y value scaled to [0,100] to make the conversion work properly. The equations are listed below.

$$[(X,Y,Z) \longrightarrow (x,y,Y)]: \begin{cases} x = \dfrac{X}{X+Y+Z} \\ y = \dfrac{Y}{X+Y+Z} \\ Y = Y \end{cases}$$

$$[Y \longrightarrow L^*]: L^* = 116\left[f\left(\frac{Y}{Y_r}\right)\right] - 16$$

$$[(X,Y,Z) \longrightarrow (L^*,u^*,v^*)]: \begin{cases} u^* = 13L^*[u' - u'_r] \\ v^* = 13L^*[v' - v'_r] \end{cases}$$

$$[(X,Y,Z) \longrightarrow (a^*,b^*)]: \begin{cases} a^* = 500\left[f\left(\dfrac{X}{X_r}\right) - f\left(\dfrac{Y}{Y_r}\right)\right] \\ b^* = 200\left[f\left(\dfrac{Y}{Y_r}\right) - f\left(\dfrac{Z}{Z_r}\right)\right] \end{cases}$$

$$[(m^*,n^*) \longrightarrow (C^*,h^*)]: \begin{cases} C^* = \sqrt{(m^*)^2 + (n^*)^2} \\ h^* = \arctan\left(\dfrac{n^*}{m^*}\right) \end{cases}$$

Where $f(t)$ is a function for LUV and LAB conversion defined by

$$f(t) = \begin{cases} 7.787t + \dfrac{16}{116} & t \le \left(\dfrac{6}{29}\right)^3 \\ (t)^{1/3} & t > \left(\dfrac{6}{29}\right)^3 \end{cases}$$

u' and v' are functions of XYZ coordinates given by

$$u' = \frac{4X}{X + 15Y + 3Z},\ v'_s = \frac{9Y}{X + 15Y + 3Z}$$

c) other unrendered color spaces to XYZ

These are the inverse transformations of part b). The equations are listed below.

$$[(x,y,Y)\longrightarrow(X,Y,Z)]:\begin{cases} X=\dfrac{xY}{y} \\ Y=Y \\ Z=\dfrac{1-x-y}{y}Y \end{cases}$$

$$[L^*\longrightarrow Y]:Y=\begin{cases}\dfrac{L^*Y_r}{903.292} & L^*\le 8\\ \left(\dfrac{L^*+16}{116}\right)^3 Y_r & L^*>8\end{cases}$$

$$[(u^*,v^*)\longrightarrow(X,Z)]:\begin{cases} Z=\dfrac{1}{3}\left[\alpha-\dfrac{\beta(\alpha+15Y)}{4}\right] \\ X=\alpha-3Z \end{cases}$$

$$[(a^*,b^*)\longrightarrow(X,Z)]:\begin{cases} X=\begin{cases}\left(f_x-\dfrac{16}{116}\right)\dfrac{X_r}{7.787} & f_x\le\left(\dfrac{216}{24389}\right)^{\frac{1}{3}}\\ f_x^3X_r & f_x>\left(\dfrac{216}{24389}\right)^{\frac{1}{3}}\end{cases} \\ Z=\begin{cases}\left(f_z-\dfrac{16}{116}\right)\dfrac{Z_r}{7.787} & f_z\le\left(\dfrac{216}{24389}\right)^{\frac{1}{3}}\\ f_z^3Z_r & f_z>\left(\dfrac{216}{24389}\right)^{\frac{1}{3}}\end{cases}\end{cases}$$

$$[(C^*,h^*)\longrightarrow(m^*,n^*)]:\begin{cases} m^*=C^*\cos(h^*) \\ n^*=C^*\sin(h^*)\end{cases}$$

Where α, β, f_x, f_y, and f_z are given by

$$\alpha=\frac{9Y}{\frac{v^*}{13L^*}+v_r^*}-15Y, \beta=\frac{u^*}{13L^*}+u_r^*$$

$$f_y=\frac{L^*+16}{116},\ f_x=\frac{a^*}{500}+f_y,\ f_z=f_y-\frac{b^*}{200}$$

d) XYZ to linear RGB and linear RGB to XYZ

Given the chromaticity coordinates of an RGB system (x_r, y_r), (x_g, y_g) and (x_b, y_b) and its reference white (X_W, Y_W, Z_W), here is the method to compute the 3×3 matrix for converting RGB to XYZ :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathrm{M}_{RGB} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Where

$$M_{RGB} = \begin{bmatrix} S_r X_r & S_g X_g & S_b X_b \\ S_r Y_r & S_g Y_g & S_b Y_b \\ S_r Z_r & S_g Z_g & S_b Z_b \end{bmatrix}$$

$$X_r = x_r / y_r$$
$$Y_r = 1$$
$$Z_r = (1 - x_r - y_r)/y_r$$
$$X_g = x_g / y_g$$
$$Y_g = 1$$
$$Z_g = (1 - x_g - y_g)/y_g$$
$$X_b = x_b / y_b$$
$$Y_b = 1$$
$$Z_b = (1 - x_b - y_b)/y_b$$

$$\begin{bmatrix} S_r \\ S_g \\ S_b \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}^{-1} \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix}$$

Use M_{RGB}^{-1} to get the RGB values from XYZ.

e) Different RGB models

Take the example where the input are XYZ coordinates (do the inverse process if input is specific RGB model), then after the linear RGB coordinates are obtained the RGB values

should be clipped to [0,1]. The target RGB coordinates in a specific RGB model needs to be gamma corrected before digitalization, which is given by

$$\text{CIE RGB Tristimulus: } \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\text{sRGB: } \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{cases} \dfrac{323(R,G,B)}{25} & (R,G,B) \leq 0.0031308 \\ \dfrac{211(R,G,B)^{5/12} - 11}{200} & (R,G,B) > 0.0031308 \end{cases}$$

$$\text{Adobe RGB: } \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}^{1/2.19921875}$$

$$\text{DCI-P3: } \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}^{1/2.6}$$

$$\text{Wide Gamut RGB: } \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}^{256/563}$$

And the final step is digitalization, given by

$$\begin{bmatrix} R'' \\ G'' \\ B'' \end{bmatrix} = \text{round}\left([2^b - 1] \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \right)$$

Where b is the bit depth.

f) Correlated color temperature and color difference

The color temperature is given by Hernandez-Andre's method, given by

$$T = \begin{cases} -949.86315 + 6253.80338e^{-\frac{n}{0.92159}} + 28.70599e^{-\frac{n}{0.20039}} + 0.00004e^{-\frac{n}{0.07125}}, & T < 50000K \\ 36284.48953 + 0.00228e^{-n/0.07861} + (5.4535 \times 10^{-36})e^{-n/0.01543} + 0.00004e^{-n/0.07125}, & T \geq 50000K \end{cases}$$

Where

$$n = \frac{x - 0.332}{y - 0.1858}$$

And x, y are the coordinates of xyY coordinates of the color.

The color difference is calculated in LAB space.

$$\Delta E^*_{ab} = \sqrt{(L^*_2 - L^*_1)^2 + (a^*_2 - a^*_1)^2 + (b^*_2 - b^*_1)^2}$$

3.2.2. Display color calibration algorithms

In the most general case, the light emitted by one pixel could be a function of the present and previous history of driving signals for the entire set of pixels on the display. It would be too difficult to characterize in color calibration. To simplify the problem, several assumptions are made in practice.

a) RGB assumption

The display is assumed to be spatially homogeneous and temporally stable. In this case, the light emitted from a pixel is only dependent on the R, G, B signal of itself, independent of other pixels' signal or the position, time. The spectral radiance of a pixel could be modified as a function of wavelength λ, signal of three channels R, G, B as

$$f(\lambda; R, G, B)$$

b) Channel-independence assumption

Another commonly used assumption is that the design and physics usually ensures that the red, green, and blue channels function independently of each other. So the spectrum can be modified further as

$$f(\lambda; R, G, B,) = f_r(\lambda, R) + f_g(\lambda, G) + f_b(\lambda, B) + f_0(\lambda)$$

Where $f_r(\lambda, R)$, $f_g(\lambda, G)$ and $f_b(\lambda, B)$ represent the spectral radiance in red, green, and blue channel respectively, and $f_0(\lambda)$ represents the back signal from a dark pixel. From this, a 3D profile is simplified to three 1D profiles.

c) Channel chromaticity constancy assumption

Additionally, the spectrum from one channel can be assumed to have a basic shape and only scaled by a factor of function of the driving signal. The mathematic expression of this assumption is given by:

$$f_r(\lambda, R) = v_r(R)s_r(\lambda)$$
$$f_g(\lambda, G) = v_g(G)s_g(\lambda)$$
$$f_b(\lambda, B) = v_b(B)s_b(\lambda)$$

Where $s(\lambda)$ is the spectrum at its maximum driving signal and $v_r(R)$, $v_g(G)$ and $v_b(B)$ represent the scaling factor. Clearly, the scaling factor is between [0,1]. The scaling factor is a function of control level and called as tone response curve (TRC).

d) Forward model

After the three assumptions, the TRCs can describe the profile of a display and used for color calibration. Considering the linearity relation between XYZ tristimulus and the corresponding spectrum, the equations above can be altered as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \begin{pmatrix} v_r(R) \\ v_g(G) \\ v_b(B) \end{pmatrix} + \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix}$$

Where we note that $T_m = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix}$ as the matrix of tristimulus values of RGB channels driven by the highest command levels, and $T_0 = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix}$ as the matrix of tristimulus values of a dark pixel.

e) Backward model

From the forward model we can get backward model:

$$\begin{pmatrix} v_r(R) \\ v_g(G) \\ v_b(B) \end{pmatrix} = T_m^{-1} \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} v_r^{-1}(v_r(R)) \\ v_g^{-1}(v_g(G)) \\ v_b^{-1}(v_b(B)) \end{pmatrix}$$

Where the inverse TRCs v_r^{-1}, v_g^{-1}, and v_b^{-1} map the real scaling factor to the correct command levels. In practice, those inverse functions problem can be solved by curve fitting or look-up table. In this project I am using the build-in curve fitting function PCHIP (Piecewise Cubic Hermite Interpolating Polynomial) in MATLAB.

3.2.3. Color space visualizer algorithms

The 3D gamut can be calculated from the locus of 2D xy chromaticity diagram with the color space conversion functions and displayed by using the MATLAB built-in function like patch, boundary and trisurf.

3.2.4. Color vision deficiency Simulation algorithms

In simulation of color appearance, it is widely used to study in LMS color space from Brettel based on human vision principle. The energy received by one kind of cones can be described as

$$[L, M, S] = \int E(\lambda)[\bar{l}, \bar{m}, \bar{s}]d\lambda$$

Where $E(\lambda)$ is the spectral power density, and $\bar{l}, \bar{m}, \bar{s}$ are spectral sensitivity of LMS cones, respectively. In the LMS space, an observable color plane is defined as

$$\alpha L + \beta M + \gamma S = 0$$

Where α, β, and γ are unknown parameters.

To solve a plane equation three reference points are needed. After converting color coordinates of RGB primaries and white point from sRGB space to LMS space, they can be used as reference. protanopia vision (Red weaknesses) and Deuteranopia vision (Green weaknesses) use origin, blue primaries, and white primaries. While Tritanopia vision (Blue weaknesses) uses red primaries rather than those of blue. Then the missing dimension can be calculated with the parameters and the other two known dimensions. For example, the new L value of protanopia vision can be calculated by

$$L_p = -(\beta M + \gamma S)/\alpha$$

From above the transformation matrix is calculated and the process of simulation is summarized as follows

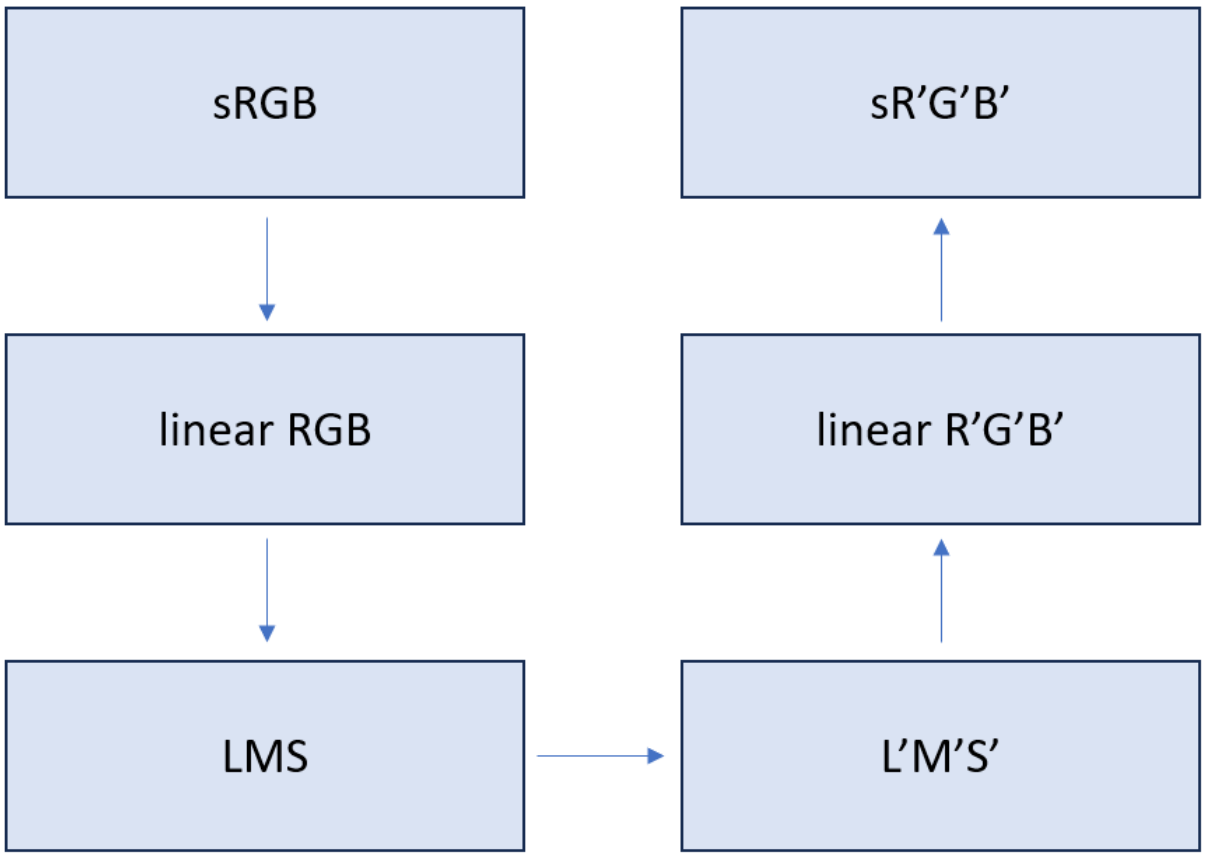

Figure 6. Dichromats vision simulation flowchart

Where the transformation between sRGB space and linear RGB space is gamma correction, and transformation between linear RGB and LMS is given by

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 17.8824 & 43.5161 & 4.11935 \\ 3.45565 & 27.1554 & 3.86714 \\ 0.0299566 & 0.184309 & 1.46709 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 0.080944 & -0.130504 & 0.116721 \\ -0.0102485 & 0.0540194 & -0.113615 \\ -0.000365294 & -0.00412163 & 0.693513 \end{bmatrix} \times \begin{bmatrix} L' \\ M' \\ S' \end{bmatrix}$$

And the equations of simulation for three types of dichromats are:

Protanopes:

$$\begin{bmatrix} L_P \\ M_P \\ S_P \end{bmatrix} = \begin{bmatrix} 0 & 2.02344 & -2.52581 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

Deuteranopes:

$$\begin{bmatrix} L_D \\ M_D \\ S_D \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.494207 & 0 & 1.24827 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

Tritanopes:

$$\begin{bmatrix} L_T \\ M_T \\ S_T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0.012245 & 0.0720345 & 0 \end{bmatrix} \times \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

After all the process, three sRGB images simulating 3 dichromats vision are obtained.

4. RESULTS and EXAMPLES

4.1. Color space conversion

The result (except CCT) can be compared with result from the Bruce Lindbloom's web calculator.

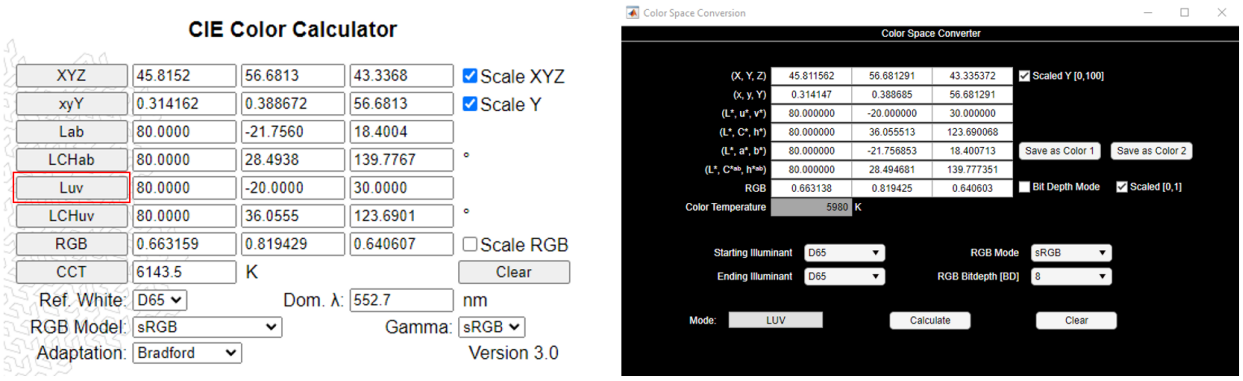

Figure 7. Comparison with Bruce Lindbloom calculator

For case study, I choose a starting color as LUV = (80, -20, 30). From the result there are some small deviations due to different precision.

4.2. Profile measurement and color calibration for monitor

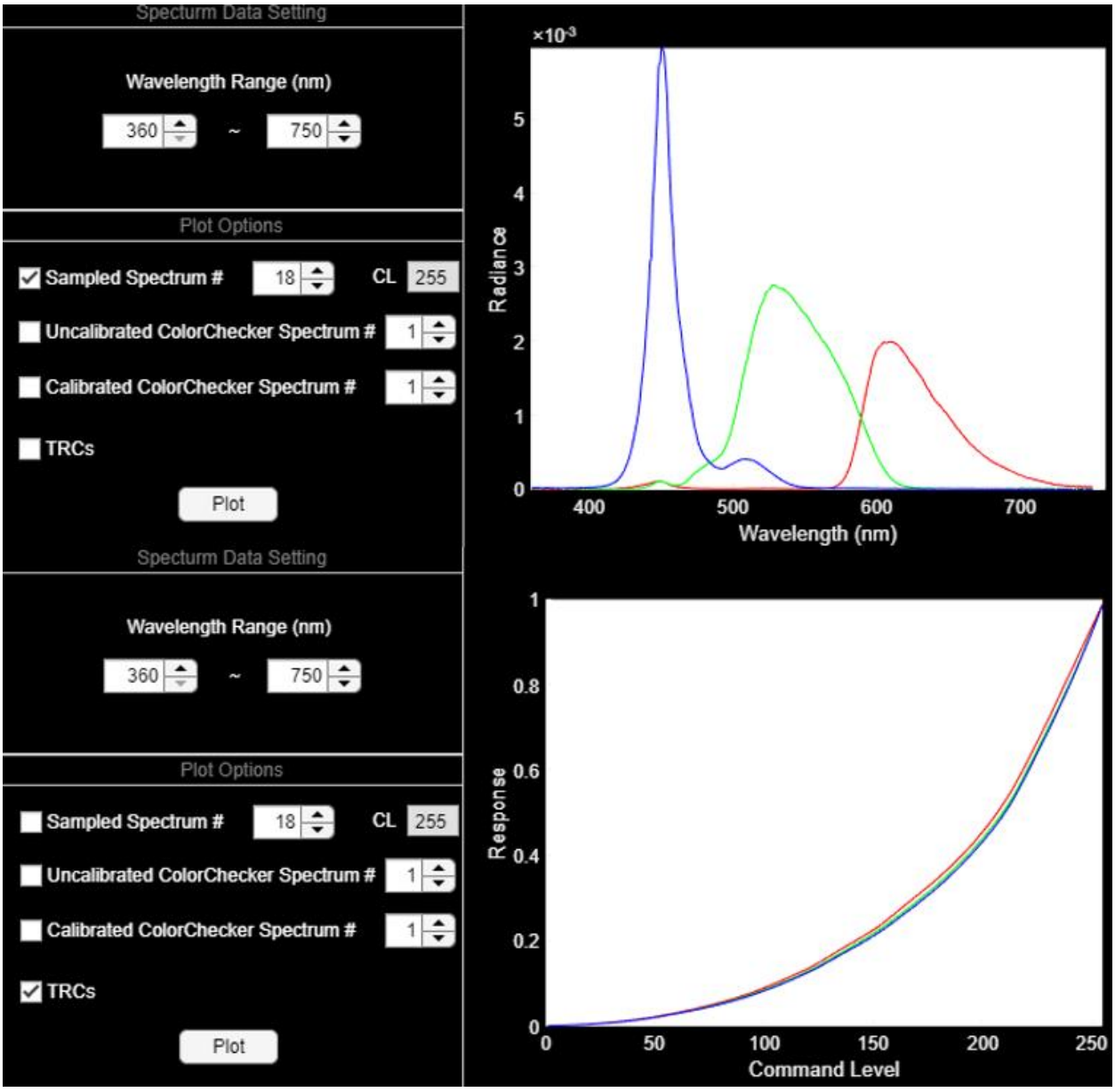

Figure 8. the spectrums of RGB channel at max command level and tone response curves

I measured the profile of an ACER monitor with SP-200 spectroradiometer from command level zero to 255 with step of 15 for every single channel. After that the TRCs and matrix were saved as excel file.

4.3. Visualization module demonstration

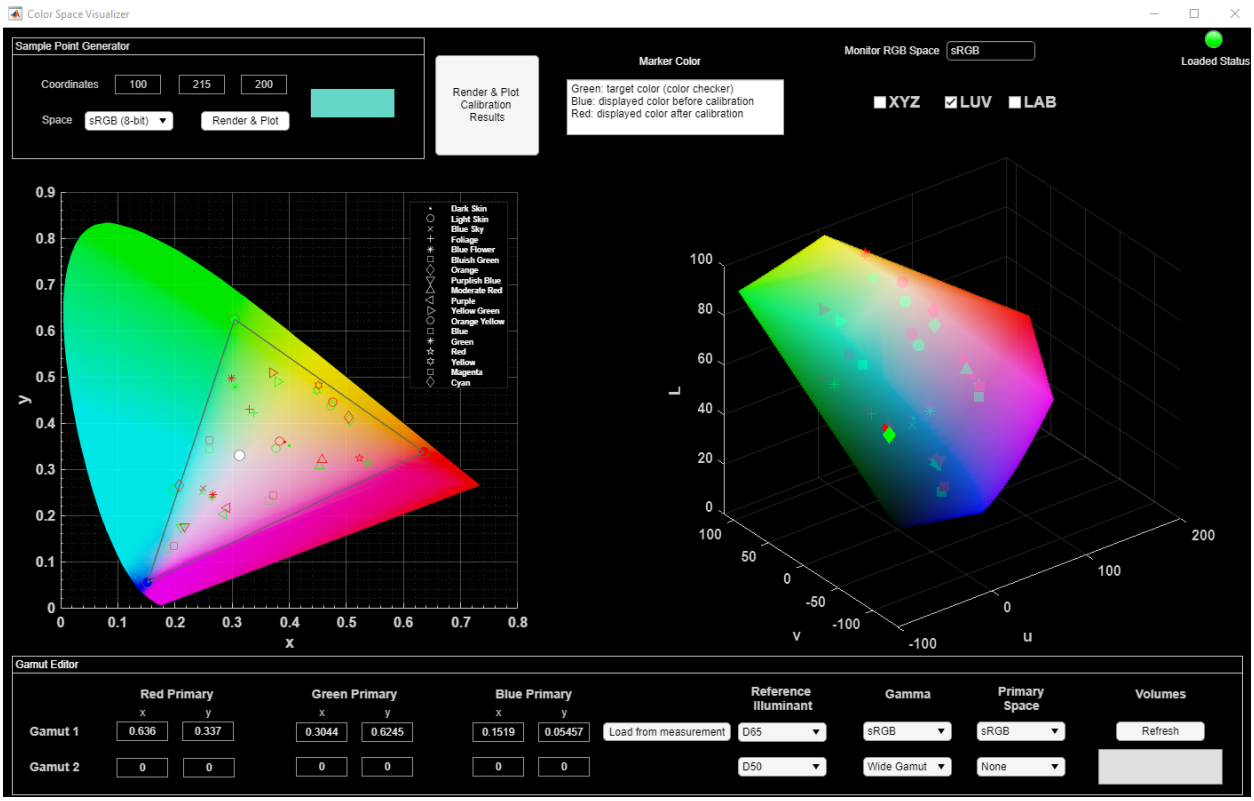

Figure 9. Calibration result in visualization module

The visualization module can also access the result from calibration module directly and plot scatters on the original plot.

4.4. Simulation module examples

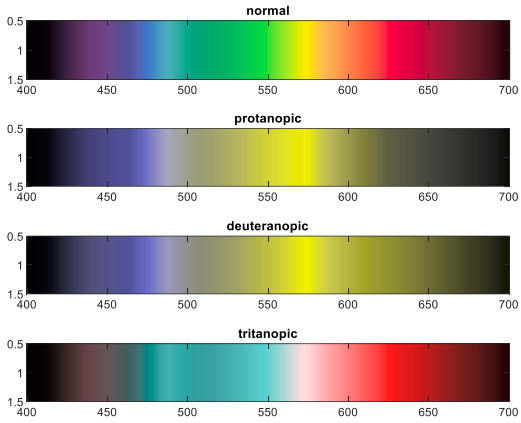

Figure 10. CVD simulation of a rainbow chart

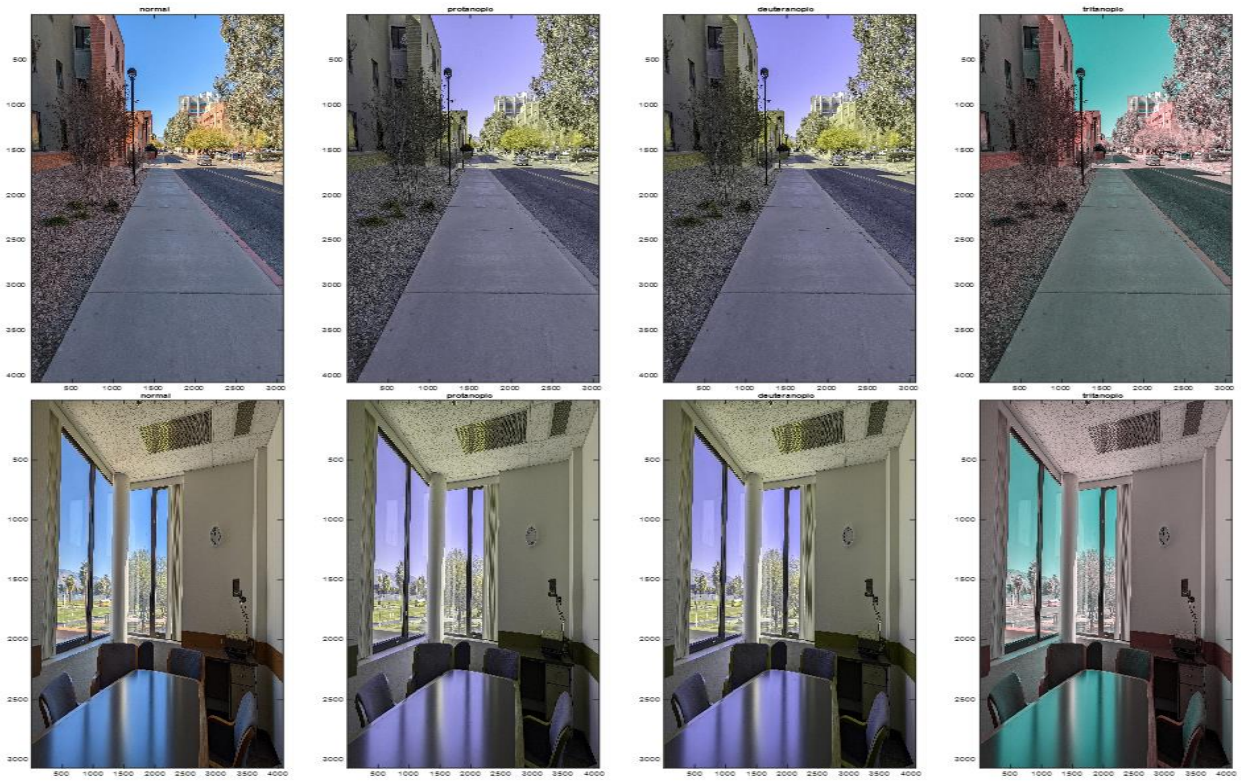

Figure 11. CVD simulation of two campus views

5. CONCLUSION

The complexities of color perception, when combined with the vast range of human visual capabilities, introduce both challenges and opportunities within digital imaging, computer vision, and design fields. This research was driven by my foundational knowledge in color science and aimed to simplify the setup for advanced experiments by crafting a MATLAB toolbox. This toolbox encompasses features for color space conversion, calibration, visualization, and simulation of color vision deficiencies.

The toolbox offers not just fundamental tools to facilitate research but also provides an avenue for individuals with normal vision to experience a glimpse of a color-blind world through CVD simulation. Moreover, this platform serves as an educational and awareness-raising tool. My journey in developing it has been enriching; I have not only broadened my understanding but also fortified my base in color science.

For the future work, there are promising prospects for enhancing the toolbox. The integration of machine learning algorithms could pave the way for automatic color calibration, drawing on environmental data or user inputs. Furthermore, introducing an enhancement method for CVD holds the potential to improve visual experiences for those who seek it.

6. APPENDIX – MATLAB Code

```
%Display Color Calibration

%denoise in frequency domain
%use truedata to denoise

%%
%close all
clear
clc

MaxCL=255;
samplenum=18;
lambda=360:750;     % Spectrum range
CL=0:MaxCL;           % Command level
%% read starting data
%SampleData=readmatrix('colorcolum2.xlsx');          % measured data of 20
sampled command Level for each RGB channel
RawData=readmatrix('Test1_SamNum18.csv','Range','B126:BC516');
WhiteData=readmatrix('Test1_SamNum18.csv','Range','BD126:BD516');
xyzbar=readmatrix('RIT_all_1nm_data.xlsx');         % xyz color matching
functions
% xyz color matching function for 360nm-750nm
xbar=xyzbar(61:451,6);
ybar=xyzbar(61:451,7);
zbar=xyzbar(61:451,8);

ColorCheckerData=readmatrix('color_checker.xlsx');  % ColorChecker 18 colors
+ white + black

%%

% Mean Background signal
BackgroundData=1/3*(RawData(:,1)+RawData(:,1+samplenum)+RawData(:,1+2*samplen
um));
% True signal before denoising
TrueData=RawData-repmat(BackgroundData,1,3*samplenum);

%% Denoise
MaskSize=20;

L=length(lambda);
x=TrueData;
% x(x<0)=0;
y=fft(x);
f=(0:(L-1)/2)/L;
p=sign(real(y)).*abs(y).^2/L;
mask=ones(size(lambda));
mask(MaskSize:end-MaskSize+1)=0;
repmask=repmat(mask',1,3*samplenum);
yy=y.*repmask;
pp=abs(yy).^2/L;
xx=ifft(yy);
```

```
FilteredData=sign(real(xx)).*abs(xx);
FilteredData(FilteredData<0)=0;

%%
% without denoise
%FilteredData=TrueData;


%%
% Tristimulus values of RGB channels
% Remove the dark signal to avoid white primaries
XYZR=[xbar ybar zbar]'*FilteredData(:,2:samplenum)*683;
XYZG=[xbar ybar zbar]'*FilteredData(:,2+samplenum:2*samplenum)*683;
XYZB=[xbar ybar zbar]'*FilteredData(:,2+2*samplenum:3*samplenum)*683;
XYZ0=[xbar ybar zbar]'*BackgroundData*683;
XYZw=[xbar ybar zbar]'*WhiteData*683;
% % Clip the negative values
% XYZR(XYZR<0)=0;
% XYZG(XYZG<0)=0;
% XYZB(XYZB<0)=0;

% Tm and T0
Tm=[XYZR(:,end),XYZG(:,end),XYZB(:,end)];

% Chromatic coordinates xy of RGB channels
Cxr=XYZR(1,:)./sum(XYZR);
Cyr=XYZR(2,:)./sum(XYZR);
Cxg=XYZG(1,:)./sum(XYZG);
Cyg=XYZG(2,:)./sum(XYZG);
Cxb=XYZB(1,:)./sum(XYZB);
Cyb=XYZB(2,:)./sum(XYZB);

% Chromatic coordinates xy of RGB channels at maximum command level
CPx=[Cxr(end),Cxg(end),Cxb(end)];
CPy=[Cyr(end),Cyg(end),Cyb(end)];

% Chromatic coordinated xy of Color Checker
CxColorChecker=ColorCheckerData(:,1);
CyColorChecker=ColorCheckerData(:,2);

% XYZ Tristimulus of Color Checker
XYZ_CC=[ColorCheckerData(:,1),ColorCheckerData(:,2),1-ColorCheckerData(:,1)-
ColorCheckerData(:,2)].*ColorCheckerData(:,3)./ColorCheckerData(:,2);
XYZ_CC=XYZ_CC';
%% Use spectral data to get TRC
% Spectral luminance 360nm-750nm of RGB channels
% Using equation from 588 Lecture#19 slide#24
fr=FilteredData(:,2:samplenum);
fg=FilteredData(:,2+samplenum:2*samplenum);
fb=FilteredData(:,2+2*samplenum:3*samplenum);
% Spectral Radiance 360nm-750nm of RGB channels at max CL
sr=FilteredData(:,samplenum);
sg=FilteredData(:,2*samplenum);
sb=FilteredData(:,3*samplenum);
% Tone response curves(TRCs)
```

```
vr=fr'*sr/sum(sr.^2);
vg=fg'*sg/sum(sg.^2);
vb=fb'*sb/sum(sb.^2);


% Clip or scale? the TRC
vr(vr>1)=1;vr(vr<0)=0;
vg(vg>1)=1;vg(vg<0)=0;
vb(vb>1)=1;vb(vb<0)=0;
vr=[0;vr];
vg=[0;vg];
vb=[0;vb];
% smooth the curve
Vr=pchip(linspace(0,MaxCL,samplenum),vr,CL);
Vg=pchip(linspace(0,MaxCL,samplenum),vg,CL);
Vb=pchip(linspace(0,MaxCL,samplenum),vb,CL);

% The piecewise polynomial structure of inverse TRCs
ppr=pchip(vr,linspace(0,MaxCL,samplenum));
ppg=pchip(vg,linspace(0,MaxCL,samplenum));
ppb=pchip(vb,linspace(0,MaxCL,samplenum));

% Calculate the control signal RGB from target XYZ (Color Checker)

RGBControl=zeros(18,3);
for i=1:18
    RGBprime=Tm\(XYZ_CC(:,i)-XYZ0);
    RGBprime(RGBprime<0)=0;RGBprime(RGBprime>255)=255;

RGBControl(i,:)=[ppval(ppr,RGBprime(1)),ppval(ppg,RGBprime(2)),ppval(ppb,RGBp
rime(3))];
end

%% Plot the chromatic diagram and mark the samples on it

figure(1)
plotChromaticity()
hold on
s=zeros(6,1);
s(1)=scatter(Cxr,Cyr,'co');
s(2)=scatter(Cxg,Cyg,'mo');
s(3)=scatter(Cxb,Cyb,'yo');
s(4)=scatter(CxColorChecker,CyColorChecker,'ks');
s(5)=scatter(CPx,CPy,'kx');
%s(6)=plot(mean([Cxr;Cxg;Cxb;Cxr],2),mean([Cyr;Cyg;Cyb;Cyr],2),'k');
s(6)=plot([CPx,CPx(1)],[CPy,CPy(1)],'k');
legend(s,'Red Channel','Green Channel','Blue
Channel','ColorChecker','Primaries at max CL')
hold off

%% Plot the TRC
figure(2)
hold on
plot(CL,Vr,'r')
plot(CL,Vg,'g')
```

```
plot(CL,Vb,'b')

title('Tone Response Curves')
ylabel('Tone Response')
xlabel('Command Level')
xlim([0 255])

%CVD Simulation Code
%Read image

ImageGT_sRGB=imread('PXL_20220204_183146991.jpg');
[ind,cmap]=rgb2ind(ImageGT_sRGB,256^2);

%%
%in pricinple we should use the spectrum of R,G,B channel at maximum
%Here I just use the table 1 from the paper

T_RGB2LMS=[17.8824 43.5161 4.11935; 3.45565 27.1554 3.86714; 0.0299566
0.184309 1.46709];
lms2lmsp = [
    0 2.02344 -2.52581;
    0 1 0;
    0 0 1];
lms2lmsd = [
    1 0 0;
    0.494207 0 1.24827;
    0 0 1];
lms2lmst = [
    1 0 0;
    0 1 0;
    -0.012245 0.0720345 0];
%RGB gamma decoding
%RGB to LMS
% SpectrumLMS=T_RGB2LMS*(imadjust(SpectrumRGB',[],[],2.2)');
ImageGT_LMS=T_RGB2LMS*(cmap'.^2.2);
% SpectrumLMS=SpectrumLMS./(T_RGB2LMS*[1;1;1]);
% SpectrumLMS(:,1:359)=NaN;
% SpectrumLMS=T_RGB2LMS*SpectrumRGB;

%LMS to LMS_Dichromat
LMS_p = lms2lmsp*ImageGT_LMS;
LMS_d = lms2lmsd*ImageGT_LMS;
LMS_t = lms2lmst*ImageGT_LMS;

%LMS_Dichromats to RGB

RGB_p=T_RGB2LMS\LMS_p;
RGB_d=T_RGB2LMS\LMS_d;
RGB_t=T_RGB2LMS\LMS_t;

RGB_p(RGB_p<0)=0;
RGB_d(RGB_d<0)=0;
RGB_t(RGB_t<0)=0;

%Gamma encoding
```

```
cmapp=RGB_p'.^(1/2.2);
cmapd=RGB_d'.^(1/2.2);
cmapt=RGB_t'.^(1/2.2);

figure(1)

tiledlayout(1,4)

nexttile
image(ind2rgb(ind,cmap));
title('normal')

nexttile
image(ind2rgb(ind,cmapp));
title('protanopic')

nexttile
image(ind2rgb(ind,cmapd));
title('deuteranopic')

nexttile
image(ind2rgb(ind,cmapt));
title('tritanopic')
```

7. REFERENCES

[1] Ibraheem, Noor A., et al. "Understanding color models: a review." *ARPN Journal of science and technology* 2.3 (2012): 265-275.

[2] Colorimetry, Schanda J. "Understanding the CIE system." *Hoboken: John Wiley & sons* (2007).

[3] Sharma, Gaurav. "LCDs versus CRTs-color-calibration and gamut considerations." *Proceedings of the IEEE* 90.4 (2002): 605-622.

[4] Vrhel, Michael J., and H. Joel Trussell. "Color device calibration: A mathematical formulation." *IEEE Transactions on Image Processing* 8.12 (1999): 1796-1806.

[5] Brettel, Hans, Françoise Viénot, and John D. Mollon. "Computerized simulation of color appearance for dichromats." *Josa a* 14.10 (1997): 2647-2655.

[6] Ruminski, J., et al. "Computerized color processing for dichromats." *Human–Computer Systems Interaction: Backgrounds and Applications 2: Part 1* (2012): 453-470.

[7] Hernandez-Andres, Javier, Raymond L. Lee, and Javier Romero. "Calculating correlated color temperatures across the entire gamut of daylight and skylight chromaticities." *Applied optics* 38.27 (1999): 5703-5709.