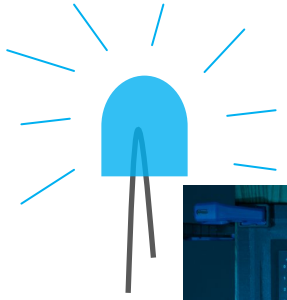
PREPROCESSING

Input image format:

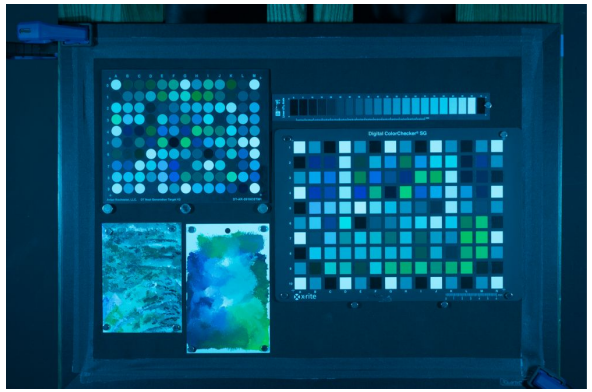
Linear RAW camera data captured in some proprietary format according to the camera manufacturer (e.g. .CR2, .NEF, .ARQ…). These image data are provided as-is, or losslessly converted prior to processing and provided as linear RAW TIFFs.

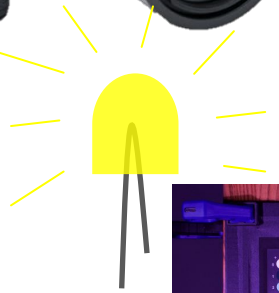
Input image set:

6 total spectral channels that come from **2 original RGB images** captured under two different lighting conditions, and provided as two RGB images.

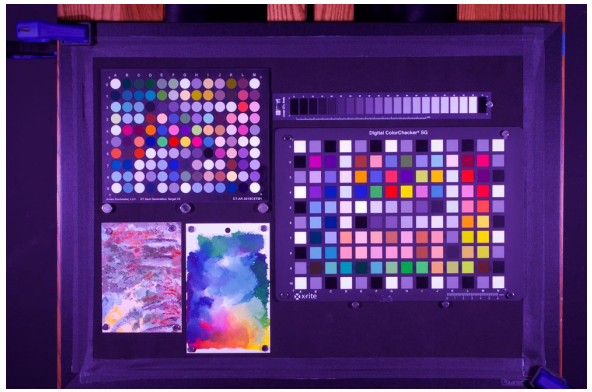


Lighting condition 1





Lighting condition 2



An RGB with 3 channels

Another RGB with 3 channels

Which are combined to give us our 6-channel spectral image stack ☺️

Bit depth scaling:

The RAW camera data is captured by the sensor at either 12 bit (4095) or 14 bit (16383) depth.

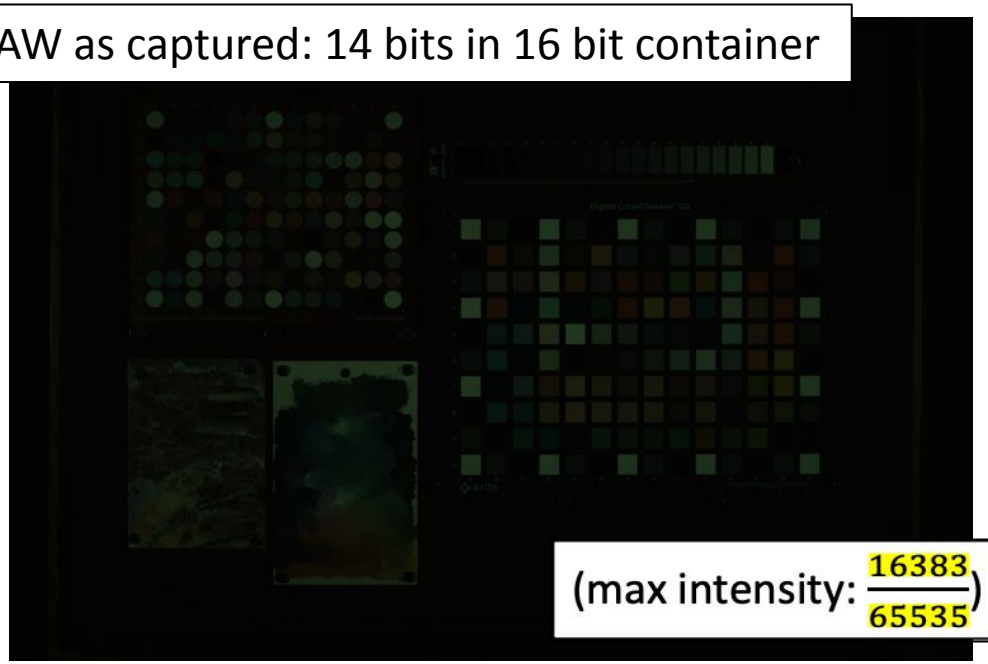
However, it is stored in an image container of 16 bit depth (65535), meaning that there are either 2 or 4 bits of unused space, causing the image to look very dark.

Therefore, we scale the RAW camera data so that the maximum for the RAW camera data matches the maximum amount of space in the 16 bit container. This fills and efficiently uses all of the space we have available.

For every pixel: $scaled\ digital\ count = RAW\ digital\ count * \frac{2^{16}-1}{2^{d}-1}$

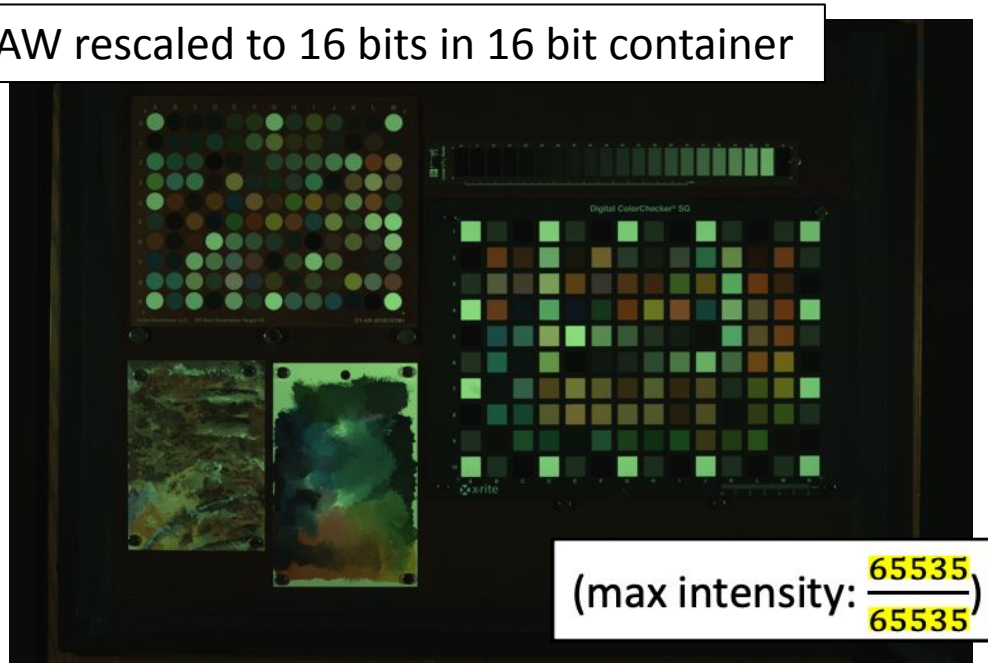
Where d is the bit depth of the RAW camera data (usually 12 or 14)*

RAW as captured: 14 bits in 16 bit container



(max intensity: $\frac{16383}{65535}$)

RAW rescaled to 16 bits in 16 bit container



(max intensity: $\frac{65535}{65535}$)

Dark current correction and Flat-fielding:

A pixel-by-pixel operation that corrects illumination and sensor nonuniformities.
For every pixel of each of the n channels:

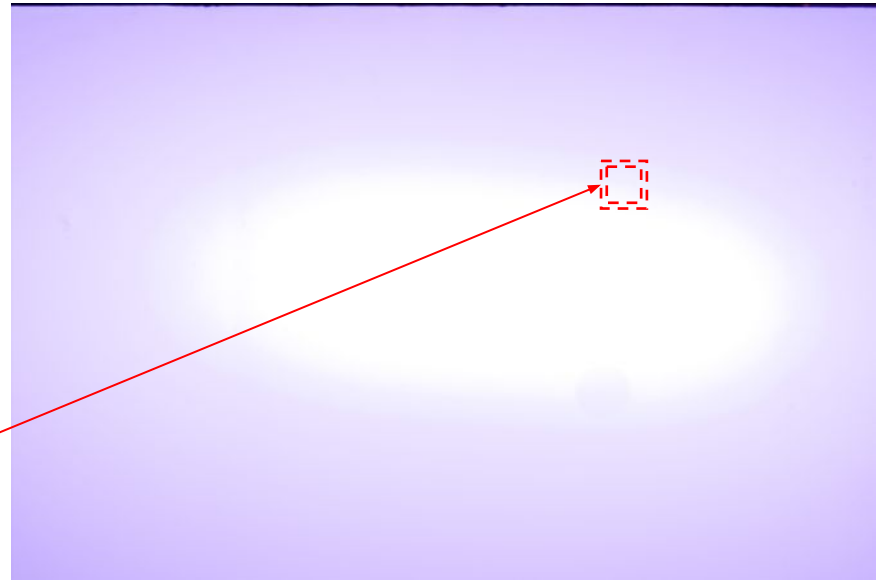
1 $$pixel_{im,n,dcc} = pixel_{im,n} - pixel_{dark,n}$$

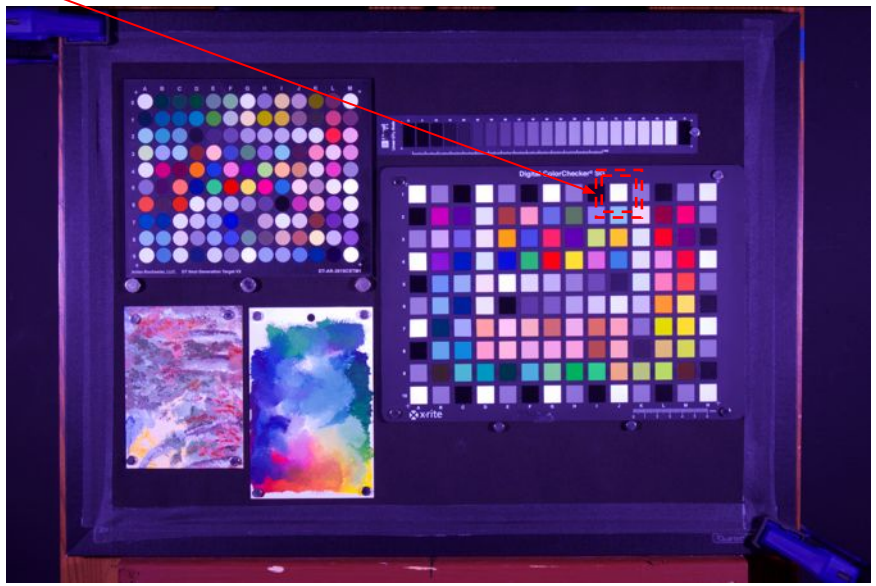
2 $$pixel_{white,n,dcc} = pixel_{white,n} - pixel_{dark,n}$$

3 $$w = Y_{white\ patch,meas} * \frac{white\ patch\ area\ average_{white,\mathbf{n=2}}}{white\ patch\ area\ average_{im,\mathbf{n=2}}}$$

Average values **AFTER** dark current correction

4 $$pixel_{im,n,ff} = w * \frac{pixel_{im,n,dcc}}{pixel_{white,n,dcc}}$$





Notes:

- im refers to the image with content in it, whether that be targets or art or both (e.g. the image to the right). It is the entity we are doing the flat-fielding to
- n identifies the spectral channel (1 through 6)
- w is calculated using **channel 2** pixel values
- **The average camera signals for the color target patches should only get extracted for calibration AFTER completing the above 4 steps!**
- To check that these steps have been implemented correctly: the value of the white patch after dark current correction and flat-fielding should be (almost) exactly equal to $Y_{white\ patch,meas}$ (e.g. for the CCSG target, $\boldsymbol{Y_{white\ patch,meas} = white\ patch\ average = 0.904989854}$).

Calculating XYZ (tristimulus values) of a reflectance sample:

$$X = k \int_{\lambda} S_\lambda R_\lambda \bar{x}_\lambda d\lambda$$

$$Y = k \int_{\lambda} S_\lambda R_\lambda \bar{y}_\lambda d\lambda$$

$$Z = k \int_{\lambda} S_\lambda R_\lambda \bar{z}_\lambda d\lambda$$

where

$$k = {100}/{\int_{\lambda} S_\lambda \bar{y}_\lambda d\lambda}$$

OR

EQUIVALENTLY…

$$X = k \sum_{\lambda} S_\lambda R_\lambda \bar{x}_\lambda \Delta\lambda$$

$$Y = k \sum_{\lambda} S_\lambda R_\lambda \bar{y}_\lambda \Delta\lambda$$

$$Z = k \sum_{\lambda} S_\lambda R_\lambda \bar{z}_\lambda \Delta\lambda$$

where

$$k = {100}/{\sum_{\lambda} S_\lambda \bar{y}_\lambda \Delta\lambda}$$

S_λ is the spectral power distribution of the illuminating light source.

R_λ is the spectral reflectance of the reflectance sample (patch on target).

$\bar{x}_{,\lambda}$, $\bar{y}_\lambda$, and $\bar{z}_\lambda$ are called Color Matching Functions (CMFs), and together, they make up a "Standard Observer." There are two different Standard Observers, differentiated by dates (1931 and 1964).

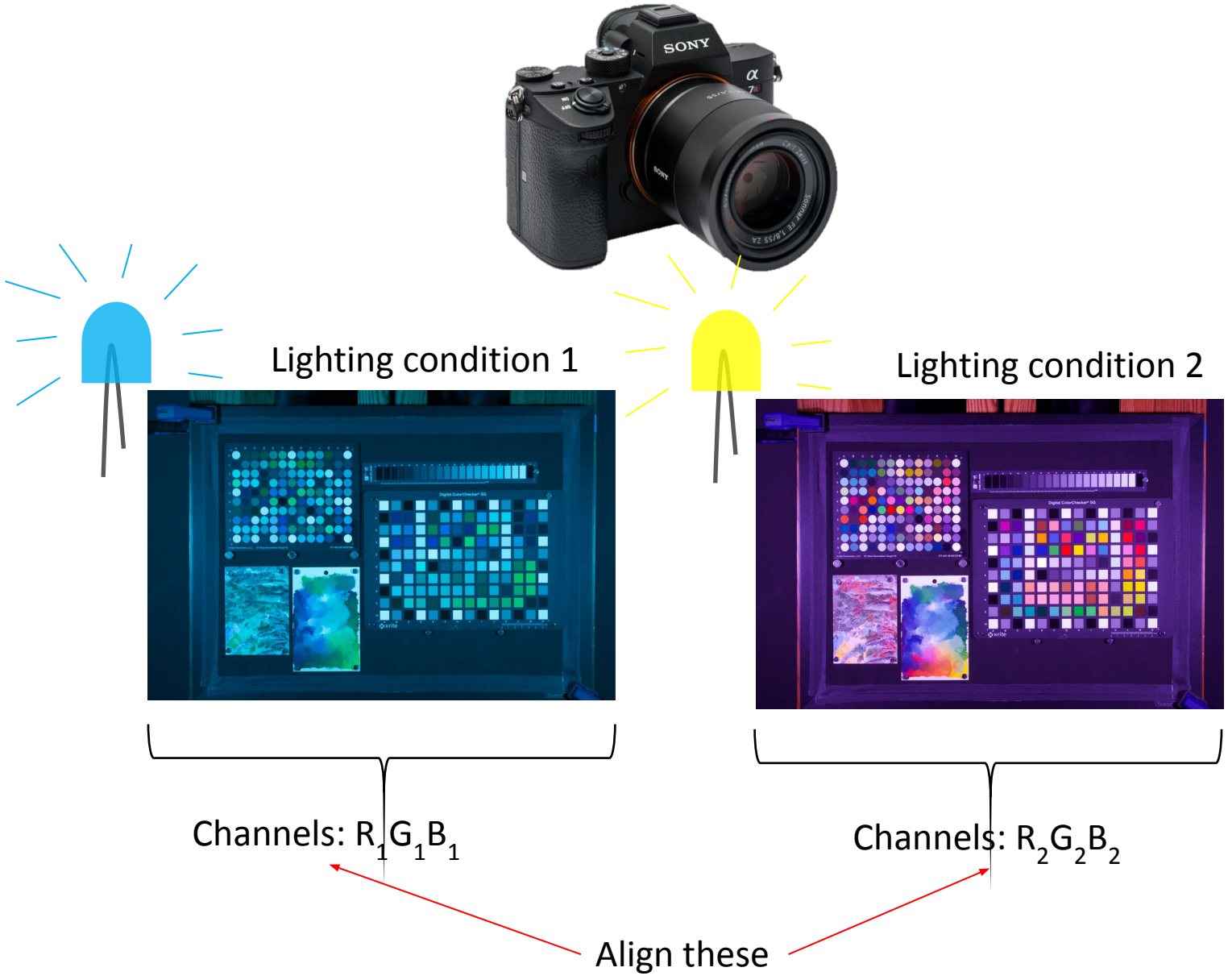
$\Delta\lambda$ is the sampling increment (*i.e.* each of the above are provided over the range of 380 nm to 780 nm, with a sampling increment of 10 nm between each data point)

See reference_data.xlsx for all of this data + a worked example calculation

Registration:

The data that the camera sensor records in each pixel will be slightly different because the light from the two different lighting conditions interacts with the optical components slightly differently. This could manifest as misalignment between the pixels of the two RGB images if you were to lay them on top of each other and compare them.

The image registration algorithm that was implemented was…



PROCESSING

XYZ to Lab:

See reference_data.xlsx

$$L^* = 116 * f\left(\frac{Y}{Y_n}\right) - 16$$

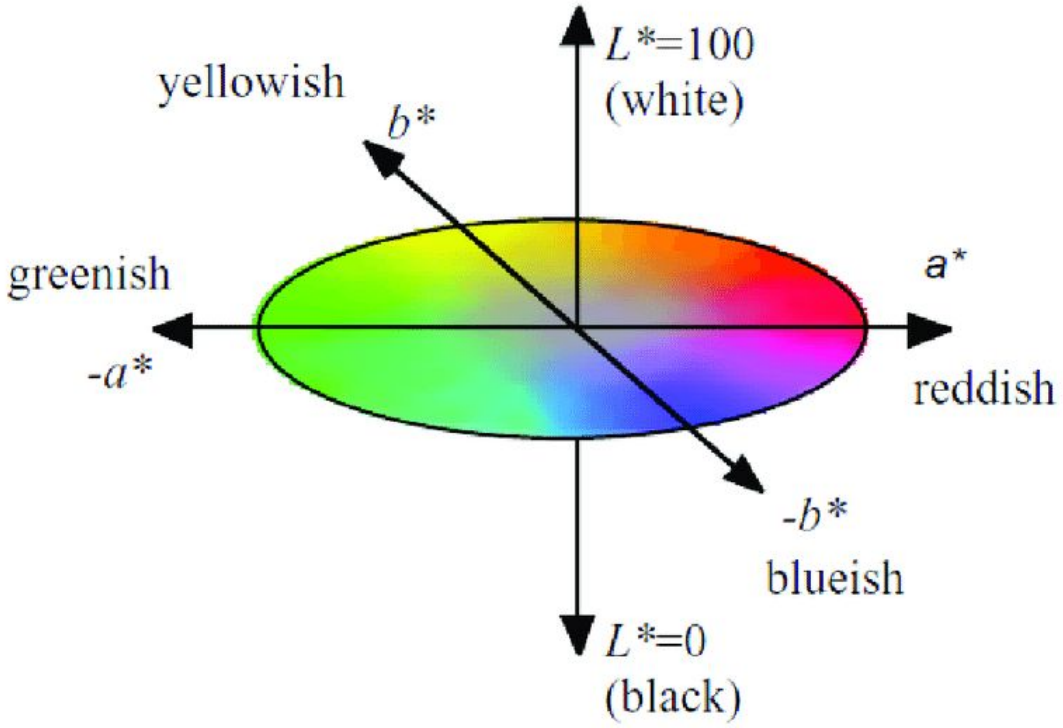
$$a^* = 500 * \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right]$$

$$b^* = 200 * \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]$$

where

$$f(x) = \begin{cases} x^{1/3}, & x > \frac{216}{24389} \\ \frac{\frac{24389}{27}x + 16}{116}, & otherwise \end{cases}$$

and X_n, Y_n, and Z_n are the XYZ values of the illuminant's white point



White Points						
Observer	1931 (2°)			1964 (10°)		
Illuminant	A	D50	D65	A	D50	D65
X_n	109.846607	96.4211994	95.0428545	111.142041	96.7206275	94.8096677
Y_n	100	100	100	100	100	100
Z_n	35.58228	82.5188285	108.890037	35.1997832	81.4280151	107.305136

CIEDE2000 Color Difference Calculation:

- **Input**: a pair of colors defined by their L*a*b* values
- **Output**: a number that quantifies how different they look to the average person
- ***Why do we have to do this?***
  - This is how we quantify the differences between the objective (measured) values of the color target patches and the values of the patches as captured by the camera
  - Then we'll do an optimization to minimize these differences, so that the imaged values are as close to the objective values as possible
    - This optimization is the "calibration" that is the meat of the processing stage

Notes:

$$k_L = 1$$
$$k_C = 1$$
$$k_H = 1$$

$$C^* = \sqrt{a^{*2} + b^{*2}}$$

$$\Delta E_{00}^* = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \frac{\Delta C'}{k_C S_C}\frac{\Delta H'}{k_H S_H}}$$

$$\Delta L' = L_2^* - L_1^*$$

$$\bar{L} = \frac{L_1^* + L_2^*}{2} \quad \bar{C} = \frac{C_1^* + C_2^*}{2}$$

$$a_1' = a_1^* + \frac{a_1^*}{2}\left(1 - \sqrt{\frac{\bar{C}^7}{\bar{C}^7 + 25^7}}\right) \quad a_2' = a_2^* + \frac{a_2^*}{2}\left(1 - \sqrt{\frac{\bar{C}^7}{\bar{C}^7 + 25^7}}\right)$$

$$\bar{C}' = \frac{C_1' + C_2'}{2} \text{ and } \Delta C' = C_2' - C_1' \quad \text{where } C_1' = \sqrt{a_1'^2 + b_1^{*2}} \quad C_2' = \sqrt{a_2'^2 + b_2^{*2}}$$

$$h_1' = \operatorname{atan2}(b_1^*, a_1') \mod 360^\circ, \quad h_2' = \operatorname{atan2}(b_2^*, a_2') \mod 360^\circ$$

$$\Delta h' = \begin{cases} h_2' - h_1' & |h_1' - h_2'| \le 180^\circ \\ h_2' - h_1' + 360^\circ & |h_1' - h_2'| > 180^\circ, h_2' \le h_1' \\ h_2' - h_1' - 360^\circ & |h_1' - h_2'| > 180^\circ, h_2' > h_1' \end{cases}$$

$$\Delta H' = 2\sqrt{C_1' C_2'}\sin(\Delta h'/2), \quad \bar{H}' = \begin{cases} (h_1' + h_2' + 360^\circ)/2 & |h_1' - h_2'| > 180^\circ \\ (h_1' + h_2')/2 & |h_1' - h_2'| \le 180^\circ \end{cases}$$

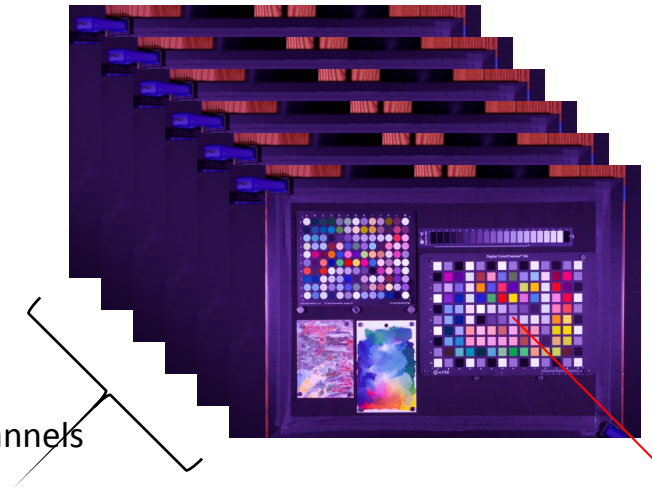
$$T = 1 - 0.17\cos(\bar{H}' - 30^\circ) + 0.24\cos(2\bar{H}') + 0.32\cos(3\bar{H}' + 6^\circ) - 0.20\cos(4\bar{H}' - 63^\circ)$$

$$S_L = 1 + \frac{0.015\left(\bar{L} - 50\right)^2}{\sqrt{20 + \left(\bar{L} - 50\right)^2}} \quad S_C = 1 + 0.045\bar{C}' \quad S_H = 1 + 0.015\bar{C}'T$$

$$R_T = -2\sqrt{\frac{\bar{C}'^7}{\bar{C}'^7 + 25^7}}\sin\left[60^\circ \cdot \exp\left(-\left[\frac{\bar{H}' - 275^\circ}{25^\circ}\right]^2\right)\right]$$

Color Transformation Matrix Optimization

1.) Setting up the matrices to calculate the XYZ values of the patches based on the ***images*** looks like this:

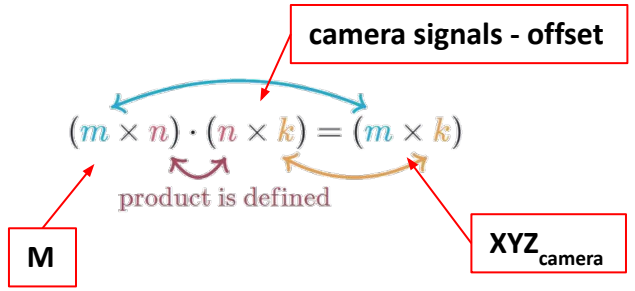


n (6) channels

average values of each patch in each channel
(i.e. the digital counts data pulled from the images using the grid)

Matrix dimensions cheat sheet:

- **M**: 3 x *n* channels (6 is our default)
- **camera signals**: *n* channels x *k* target patches
- **XYZ$_{camera}$**: 3 x *k* patches



$$\boldsymbol{XYZ_{camera} = M * (camera\ signals - offset)}$$

which expanded looks like…

$$\begin{bmatrix} X_{patch\,1} & X_{patch\,2} & \cdots & X_{patch\,k} \\ Y_{patch\,1} & Y_{patch\,2} & \cdots & Y_{patch\,k} \\ Z_{patch\,1} & Z_{patch\,2} & \cdots & Z_{patch\,k} \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} & m_{1,6} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} & m_{2,6} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} & m_{3,6} \end{bmatrix} * \begin{bmatrix} camsigs_{ch\,1,patch\,1} - o_1 & \cdots & camsigs_{ch\,1,patch\,k} - o_1 \\ camsigs_{ch\,2,patch\,1} - o_2 & \cdots & camsigs_{ch\,2,patch\,k} - o_2 \\ camsigs_{ch\,3,patch\,1} - o_3 & \cdots & camsigs_{ch\,3,patch\,k} - o_3 \\ camsigs_{ch\,4,patch\,1} - o_4 & \cdots & camsigs_{ch\,4,patch\,k} - o_4 \\ camsigs_{ch\,5,patch\,1} - o_5 & \cdots & camsigs_{ch\,5,patch\,k} - o_5 \\ camsigs_{ch\,6,patch\,1} - o_6 & & camsigs_{ch\,6,patch\,k} - o_6 \end{bmatrix}$$

M and *offset are* the matrices that will be optimized to minimize the color difference between the measured patch colors (reference_data.xlsx) and the imaged patch colors

Color Transformation Matrix Optimization (cont.)

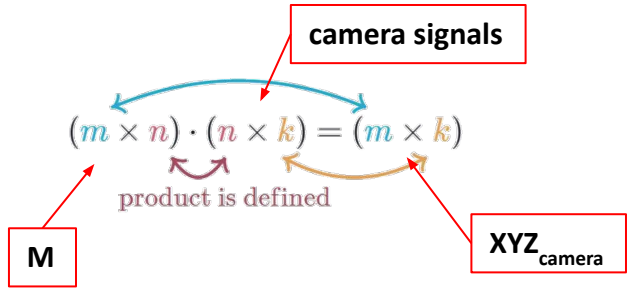
1a.) Starting the optimization process requires providing starting values with which to populate *M* and *offset*:

$$M_{initial} = \begin{bmatrix} 0.1 & 0.1 & 0.25 & 0.5 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.25 & 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 0.25 & 0.1 & 0.1 & 0.5 \end{bmatrix}$$

$$offset_{initial} = [\,0.01 \quad 0.01 \quad 0.01 \quad 0.01 \quad 0.01 \quad 0.01\,]'$$

Matrix dimensions cheat sheet:

- **M**: 3 x *n* channels (6 is our default)
- **camera signals**: *n* channels x *k* target patches
- **XYZ$_{camera}$**: 3 x *k* patches



$$\boldsymbol{XYZ_{camera} = M * (camera\ signals - offset)}$$

which expanded looks like…

$$\begin{bmatrix} X_{patch\,1} & X_{patch\,2} & \cdots & X_{patch\,k} \\ Y_{patch\,1} & Y_{patch\,2} & \cdots & Y_{patch\,k} \\ Z_{patch\,1} & Z_{patch\,2} & \cdots & Z_{patch\,k} \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} & m_{1,6} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} & m_{2,6} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} & m_{3,6} \end{bmatrix} * \begin{bmatrix} camsigs_{ch\,1,patch\,1} - o_1 & \cdots & camsigs_{ch\,1,patch\,k} - o_1 \\ camsigs_{ch\,2,patch\,1} - o_2 & \cdots & camsigs_{ch\,2,patch\,k} - o_2 \\ camsigs_{ch\,3,patch\,1} - o_3 & \cdots & camsigs_{ch\,3,patch\,k} - o_3 \\ camsigs_{ch\,4,patch\,1} - o_4 & \cdots & camsigs_{ch\,4,patch\,k} - o_4 \\ camsigs_{ch\,5,patch\,1} - o_5 & \cdots & camsigs_{ch\,5,patch\,k} - o_5 \\ camsigs_{ch\,6,patch\,1} - o_6 & & camsigs_{ch\,6,patch\,k} - o_6 \end{bmatrix}$$

Color Transformation Matrix Optimization (cont.)

2.) Iteratively change the coefficients in *M* until the average color difference (ΔE_{00}) between the colors of all of the patches as captured by the camera (XYZ_{camera} → $L^*a^*b^*_{camera}$) and the ground truth measurements ($L^*a^*b^*_{reference}$) is minimized

OPTIMIZATION

CAMERA

1.) Plug some values into these two matrices

2.) Calculate this

$$XYZ_{camera} = M * (camera\ signals - offset)$$

(XYZ to $L^*a^*b^*$ functions)

3.) Then calculate this: $L^*a^*b^*_{camera}$

REALITY

Reference target data (reference_data.xlsx)

OR

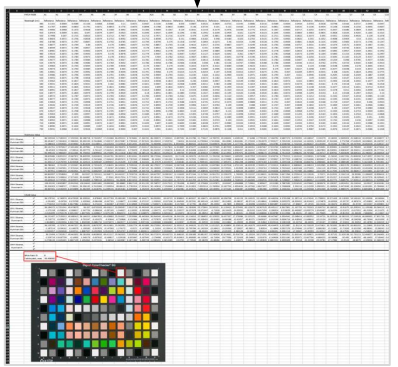
Measurements provided by user

Gives

$XYZ_{reference}$

and

$L^*a^*b^*_{reference}$



CIEDE2000 formula

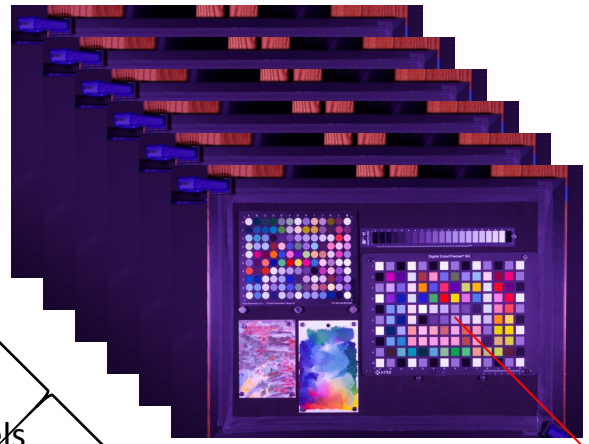
Minimum **mean** ΔE_{00}?

no → Try new coefficients in *M* and *offset* and repeat

yes → Yay, your matrix is optimized! You're done!

Spectral Transformation Matrix Optimization

1.) Setting up the matrices to calculate the spectral reflectance of the patches based on the ***images*** looks like this:



n (6) channels

average values of each patch in each channel
(i.e. the digital counts data pulled from the images using the grid)

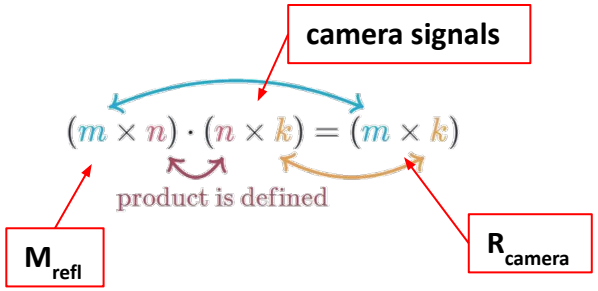
$$\boldsymbol{R_{camera} = M_{refl} * camera\ signals}$$

which expanded looks like…

$$\begin{bmatrix} R_{\lambda_1,1} & R_{\lambda_1,2} & \cdots & R_{\lambda_1,k} \\ R_{\lambda_2,1} & R_{\lambda_2,2} & \cdots & R_{\lambda_2,k} \\ \vdots & \vdots & \cdots & \vdots \\ R_{\lambda_{36},1} & R_{\lambda_{36},2} & \cdots & R_{\lambda_{36},k} \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} & m_{1,6} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} & m_{2,6} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{36,1} & m_{36,2} & m_{36,3} & m_{36,4} & m_{36,5} & m_{36,6} \end{bmatrix} * \begin{bmatrix} camsigs_{ch\ 1, patch\ 1} & & camsigs_{ch\ 1, patch\ k} \\ camsigs_{ch\ 2, patch\ 1} & \cdots & camsigs_{ch\ 2, patch\ k} \\ camsigs_{ch\ 3, patch\ 1} & \vdots & camsigs_{ch\ 3, patch\ k} \\ camsigs_{ch\ 4, patch\ 1} & \vdots & camsigs_{ch\ 4, patch\ k} \\ camsigs_{ch\ 5, patch\ 1} & \cdots & camsigs_{ch\ 5, patch\ k} \\ camsigs_{ch\ 6, patch\ 1} & & camsigs_{ch\ 6, patch\ k} \end{bmatrix}$$

Matrix dimensions cheat sheet:

- **M**: 36 x *n* channels (6 is our default)
- **camera signals**: *n* channels x *k* target patches
- $\mathbf{R}_{\mathbf{camera}}$: 36 x *k* patches



M_{refl} is the matrix that will be optimized to minimize the RMS error between the measured patch spectral reflectances (reference_data.xlsx) and the imaged patch spectral reflectances

Spectral Transformation Matrix Optimization

1a.) Starting the optimization process requires providing starting values with which to populate M_{refl}:

$$M_{refl,initial} = R_{reference} * pseudoinverse(camera\ signals)$$

From reference_data.xlsx

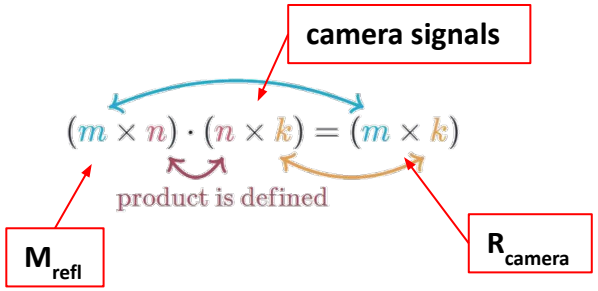
$$\boldsymbol{R_{camera} = M_{refl} * camera\ signals}$$

which expanded looks like…

$$\begin{bmatrix} R_{\lambda_1,1} & R_{\lambda_1,2} & \cdots & R_{\lambda_1,k} \\ R_{\lambda_2,1} & R_{\lambda_2,2} & \cdots & R_{\lambda_2,k} \\ \vdots & \vdots & \cdots & \vdots \\ R_{\lambda_{36},1} & R_{\lambda_{36},2} & \cdots & R_{\lambda_{36},k} \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} & m_{1,6} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} & m_{2,6} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{36,1} & m_{36,2} & m_{36,3} & m_{36,4} & m_{36,5} & m_{36,6} \end{bmatrix} * \begin{bmatrix} camsigs_{ch\ 1,patch\ 1} & & camsigs_{ch\ 1,patch\ k} \\ camsigs_{ch\ 2,\ patch\ 1} & \cdots & camsigs_{ch\ 2,\ patch\ k} \\ camsigs_{ch\ 3,\ patch\ 1} & \vdots & camsigs_{ch\ 3,\ patch\ k} \\ camsigs_{ch\ 4,\ patch\ 1} & \vdots & camsigs_{ch\ 4,\ patch\ k} \\ camsigs_{ch\ 5,\ patch\ 1} & \cdots & camsigs_{ch\ 5,\ patch\ k} \\ camsigs_{ch\ 6,\ patch\ 1} & & camsigs_{ch\ 6,\ patch\ k} \end{bmatrix}$$

Matrix dimensions cheat sheet:

- **M**: 36 x *n* channels (6 is our default)
- **camera signals**: *n* channels x *k* target patches
- $\mathbf{R_{camera}}$: 36 x *k* patches



Spectral Transformation Matrix Optimization (cont.)

2.) Iteratively change the coefficients in M_{refl} until the weighted error term *Z*, which sums RMS/SS error differences between the reference reflectance spectra and the reflectance spectra estimated from the camera data, is minimized

OPTIMIZATION

CAMERA

1.) Plug some values into this matrix

2.) Calculate this

$$R_{camera} = M_{refl} * (camera\ signals)$$

and find

$R_{camera,global\ max}, R_{camera,global\ min}$
(at every wavelength)

3.) Calculate these error terms

$$Z = e_1 + 10e_2 + 50e_3$$

where

$$e_1 = \sqrt{\sum\sum (R_{reference} - R_{camera})^2}$$

$$e_2 = \sum (R_{camera,global\ max} - R_{reference,global\ max})^2$$

$$e_3 = \sum (R_{camera,global\ min} - R_{reference,global\ min})^2$$

REALITY

Reference target data (reference_data.xlsx)

OR

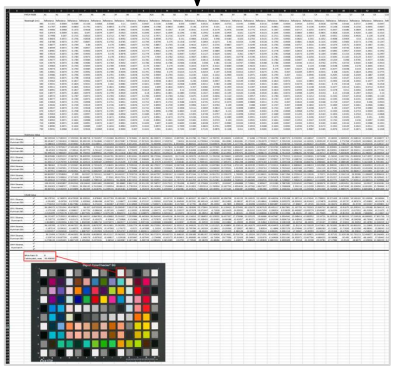
Measurements provided by user

Gives

$R_{reference}$

find

$R_{reference,global\ max}, R_{reference,global\ min}$
(at every wavelength)



Minimum ***Z***?

no

Try new coefficients in M_{refl} and repeat

yes

Yay, your matrix is optimized! You're done!

RENDERING

Computing the Color-Calibrated Image

1.) Apply the **optimized** *M* and *offset* to every pixel of every channel to compute the preliminary XYZ version of the final image:

$$\boldsymbol{XYZ_{camera} = M * (camera\ signals - offset)}$$

which expands to…

$$\begin{bmatrix} X_{pixel\ 1} & X_{pixel\ 2} & \cdots & X_{pixel\ k} \\ Y_{pixel\ 1} & Y_{pixel\ 2} & \cdots & Y_{pixel\ k} \\ Z_{ixel\ 1} & Z_{pixel\ 2} & \cdots & Z_{pixel\ k} \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} & m_{1,6} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} & m_{2,6} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} & m_{3,6} \end{bmatrix} * \begin{bmatrix} cam\ sig_{ch\ 1,pixel\ 1} - o_1 & & cam\ sig_{ch\ 1,\ pixel\ k} - o_1 \\ cam\ sig_{ch\ 2,\ pixel\ 1} - o_2 & \cdots & cam\ sig_{ch\ 2,\ pixel\ k} - o_2 \\ cam\ sig_{ch\ 3,\ pixel\ 1} - o_3 & \vdots & cam\ sig_{ch\ 3,\ pixel\ k} - o_3 \\ cam\ sig_{ch\ 4,\ pixel\ 1} - o_4 & \vdots & cam\ sig_{ch\ 4,\ pixel\ k} - o_4 \\ cam\ sig_{ch\ 5,\ pixel\ 1} - o_5 & \cdots & cam\ sig_{ch\ 5,\ pixel\ k} - o_5 \\ cam\ sig_{ch\ 6,\ patch\ 1} - o_6 & & cam\ sig_{ch\ 6,\ pixel\ k} - o_6 \end{bmatrix}$$

…and will give a 3-by-(image rows x image cols) matrix of the XYZ values of every pixel.

2.) Convert to linear ProPhoto color space using the XYZ to RGB matrix (M_{PP}) provided below:

$$\boldsymbol{RGB_{PP,\ linear} = M * XYZ_{camera}}$$

$$M = \begin{bmatrix} 1.3459433 & -0.2556075 & -0.0511118 \\ -0.5445989 & 1.5081673 & 0.0205351 \\ 0.0000000 & 0.0000000 & 1.2118128 \end{bmatrix}$$

Computing the Color-Calibrated Image

3.) Clip the linear RGB values between 0 and 1:

$$RGB_{PP,\,linear,\,clipped} = f(RGB_{PP,\,linear})$$

$$f(x) = \begin{cases} 0\,, & x < 0 \\ x\,, & 0 \le x \le 1 \\ 1\,, & x > 1 \end{cases}$$

4.) Apply gamma:

$$RGB_{PP,final} = f(RGB_{PP,\,linear,clipped})$$

$$f(x) = \begin{cases} x^{1/1.8}\,, & x \ge 0.001953125 \\ x * 16\,, & otherwise \end{cases}$$

Going from $RGB_{ProPhoto}$ to XYZ to RGB_{sRGB}

1.) Re-linearize the RGB_{PP} values by applying the inverse gamma:

$$\boldsymbol{RGB_{PP,linear} = f(RGB_{PP,final})}$$

$$f(x) = \begin{cases} x^{1.8}, & x \geq 16*0.001953125 \\ \dfrac{x}{16}, & otherwise \end{cases}$$

2.) Convert from linear ProPhoto RGB values back to XYZ values by applying the inverse transform (M_{PP}^{-1}) provided below:

$$\boldsymbol{XYZ = M^{-1} * RGB_{PP,\,linear}}$$

$$M^{-1} = \begin{bmatrix} 0.7976749 & 0.1351917 & 0.0313534 \\ 0.2880402 & 0.7118741 & 0.0000857 \\ 0.0000000 & 0.0000000 & 0.8252100 \end{bmatrix}$$

3.) Convert to linear sRGB color space using the XYZ to RGB matrix (M_{sRGB}) provided below:

$$\boldsymbol{RGB_{sRGB,\,linear} = M_{sRGB} * XYZ}$$

$$M_{sRGB} = \begin{bmatrix} 3.1338561 & -1.6168667 & -0.4906146 \\ -0.9787684 & 1.9161415 & 0.0334540 \\ 0.0719453 & -0.2289914 & 1.4052427 \end{bmatrix}$$

4.) Clip the linear sRGB values between 0 and 1:

$$\boldsymbol{RGB_{sRGB,\,linear,\,clipped} = f(RGB_{sRGB,\,linear})}$$

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases}$$

5.) Apply the sRGB gamma:

$$\boldsymbol{RGB_{sRGB,final} = f(RGB_{sRGB,\,linear,clipped})}$$

$$f(x) = \begin{cases} 1.055 * x^{1/2.4} - 0.055, & x > 0.0031308 \\ x * 12.92, & otherwise \end{cases}$$

VERIFICATION

Verifying the Accuracy of the Color Transformation Matrix

VERIFICATION

CAMERA

1.) Plug the optimized *M* and *offset* matrices in

2.) Calculate this

$$XYZ_{camera} = M * (camera\ signals - offset)$$

(XYZ to L*a*b* functions)

1a.) Plug the camera signals from the second target in the image in here

3.) Then calculate this: $L^*a^*b^*_{camera}$

4.) Then this: CIEDE2000 formula

REALITY

Reference target data (reference_data.xlsx)

OR

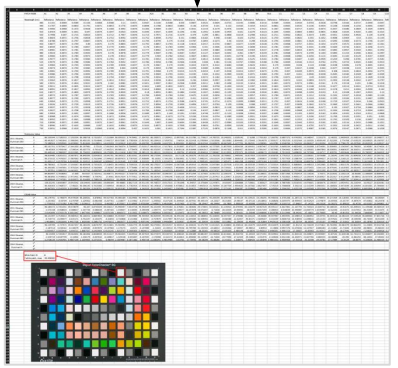
Measurements provided by user

Gives

$XYZ_{reference}$

and

$L^*a^*b^*_{reference}$



Save XYZ_{camera}, $L^*a^*b^*_{camera}$, and the ΔE_{00} values for exporting/reporting in the viewer with the calibration results

Verifying the Accuracy of the Spectral Transformation Matrix

VERIFICATION

CAMERA

REALITY

1.) Plug the optimized M_{refl} in here

2.) Calculate this

$$R_{camera} = M_{refl} * (camera\ signals)$$

1a.) Plug the camera signals from the second target in the image in here

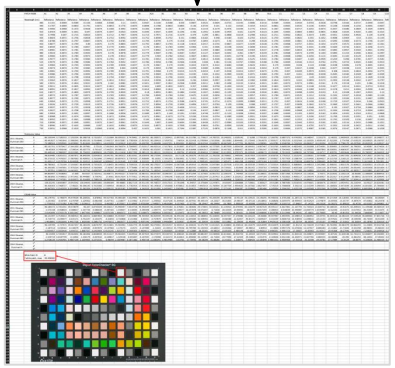
Reference target data (reference_data.xlsx)

OR

Measurements provided by user

Gives

$R_{reference}$



3.) Then calculate RMSE:

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(R_{camera,i} - R_{reference,i})^2}{N}}$$

Save R_{camera} and RMSE for exporting/reporting in the viewer with the calibration results

SPECTRAL PICKING

Computing Estimated Reflectance Spectra in the Spectral Picker:

1.) Average the digital counts for each of the 6 channels from all the pixels in the selected region (this will give a 6-by-1 matrix of camera signal values - see below).

2.) Apply the **optimized** M_{refl} to the averaged camera signals:

$$\boldsymbol{R_{camera} = M_{refl} * camera\ signals}$$

which expands to…

$$\begin{bmatrix} R_{camera\,\lambda_1} \\ R_{camera\,\lambda_2} \\ \vdots \\ R_{camera\,\lambda_{36}} \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} & m_{1,6} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} & m_{2,6} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{36,1} & m_{36,2} & m_{36,3} & m_{36,4} & m_{36,5} & m_{36,6} \end{bmatrix} * \begin{bmatrix} cam\ sig_{ch\ 1\ average} \\ cam\ sig_{ch\ 2\ average} \\ cam\ sig_{ch\ 3\ average} \\ cam\ sig_{ch\ 4\ average} \\ cam\ sig_{ch\ 5,\ average} \\ cam\ sig_{ch\ 6\ average} \end{bmatrix}$$

…and will give a 36-by-1 matrix of the reflectance spectrum of the averaged pixels.