IRL #11: Progress Review
Man-Ning Chen (Mandy)
Team G: EXCALIBR

Teammates: Yiqing Cai, Huan-Yang Chang,

Siddharth Raina, Sambuddha Sarkar

Individual Progress

Color Calibration

Color calibration is to measure and adjust the color response of a device (input or output) to a known state. We use an X-Rite ColorChecker Classic Card as our ground truth (The manufacture gives us the true color values) and aim at mapping the colors recorded by the cameras these ground truths.

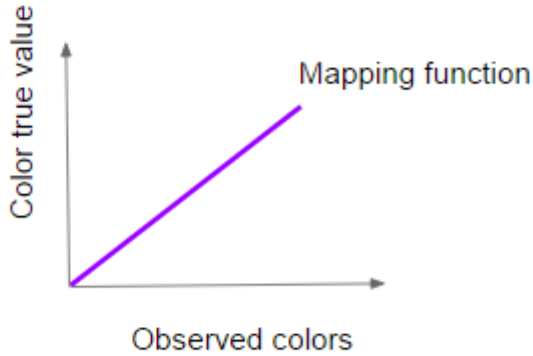


Figure 1. Mapping function (unreal): This graph is only for illustrating the concepts

Current Status

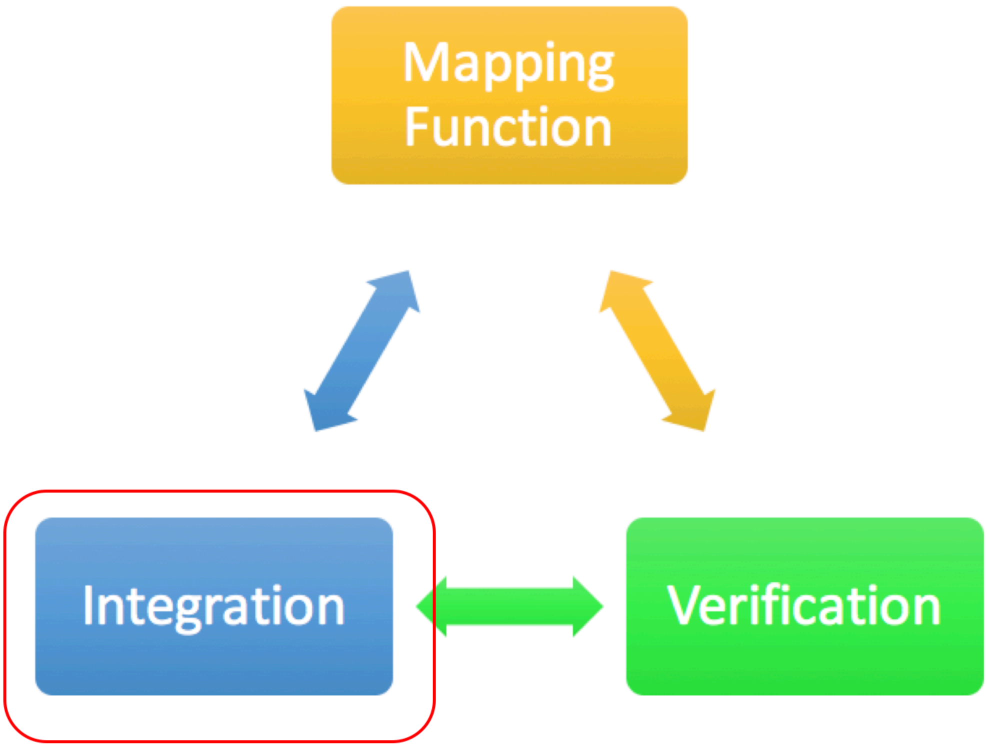


Figure 2. Status

For PR 12, we focus on Since the whole system is still under construction, EVT color camera is not available for me to test this week. However, in order to perform integration, I tested my algorithm in our future working environment. This environment can provide special lighting condition and color calibration helps to fix the special lighting.

ColorChecker Information

Figure 3. The 1-24 in this image correspond to the 1-24 in the x-axis below

No.	Number		sRGB R	sRGB G	sRGB B	CIE L*a*b* L*	CIE L*a*b* a*	CIE L*a*b* b*	Munsell Notation Hue	Munsell Notation Value / Chroma
1.	dark skin		115	82	68	37.986	13.555	14.059	3 YR	3.7 / 3.2
2.	light skin		194	150	130	65.711	18.13	17.81	2.2 YR	6.47 / 4.1
3.	blue sky		98	122	157	49.927	-4.88	-21.925	4.3 PB	4.95 / 5.5
4.	foliage		87	108	67	43.139	-13.095	21.905	6.7 GY	4.2 / 4.1
5.	blue flower		133	128	177	55.112	8.844	-25.399	9.7 PB	5.47 / 6.7
6.	bluish green		103	189	170	70.719	-33.397	-0.199	2.5 BG	7 / 6
7.	orange		214	126	44	62.661	36.067	57.096	5 YR	6 / 11
8.	purplish blue		80	91	166	40.02	10.41	-45.964	7.5 PB	4 / 10.7
9.	moderate red		193	90	99	51.124	48.239	16.248	2.5 R	5 / 10
10.	purple		94	60	108	30.325	22.976	-21.587	5 P	3 / 7
11.	yellow green		157	188	64	72.532	-23.709	57.255	5 GY	7.1 / 9.1
12.	orange yellow		224	163	46	71.941	19.363	67.857	10 YR	7 / 10.5
13.	blue		56	61	150	28.778	14.179	-50.297	7.5 PB	2.9 / 12.7
14.	green		70	148	73	55.261	-38.342	31.37	0.25 G	5.4 / 8.65
15.	red		175	54	60	42.101	53.378	28.19	5 R	4 / 12
16.	yellow		231	199	31	81.733	4.039	79.819	5 Y	8 / 11.1
17.	magenta		187	86	149	51.935	49.986	-14.574	2.5 RP	5 / 12
18.	cyan		8	133	161	51.038	-28.631	-28.638	5 B	5 / 8
19.	white (.05*)		243	243	242	96.539	-0.425	1.186	N	9.5 /
20.	neutral 8 (.23*)		200	200	200	81.257	-0.638	-0.335	N	8 /
21.	neutral 6.5 (.44*)		160	160	160	66.766	-0.734	-0.504	N	6.5 /
22.	neutral 5 (.70*)		122	122	121	50.867	-0.153	-0.27	N	5 /
23.	neutral 3.5 (.1.05*)		85	85	85	35.656	-0.421	-1.231	N	3.5 /
24.	black (1.50*)		52	52	52	20.461	-0.079	-0.973	N	2 /

Cie L*a*b* values use Illuminant D50 2 degree observer sRGB values for Illuminate D65.

Figure 3. Color ground truth

Mapping function

The mapping function currently used:

Polynomial Model:

$$CalibratedR = \alpha_0 + \alpha_1 R + \alpha_2 G + \alpha_3 B + \alpha_4 R^2 + \alpha_5 G^2 + \alpha_6 B^2$$

$$CalibratedG = \beta_0 + \beta_1 R + \beta_2 G + \beta_3 B + \beta_4 R^2 + \beta_5 G^2 + \beta_6 B^2$$

$$CalibratedB = \gamma_0 + \gamma_1 R + \gamma_2 G + \gamma_3 B + \gamma_4 R^2 + \gamma_5 G^2 + \gamma_6 B^2$$

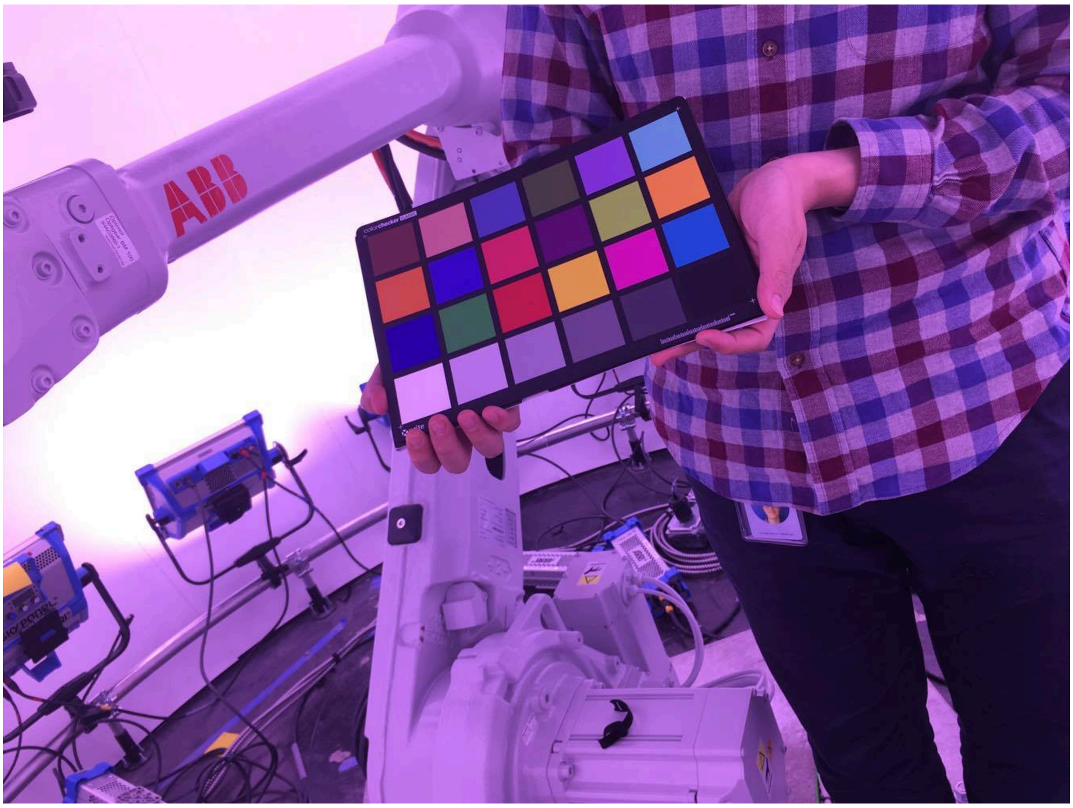
Fitting Method:

$$\begin{pmatrix} B^2 & G^2 & R^2 & B & G & C & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}_{Camera} \begin{bmatrix} \alpha_6 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} R \\ \vdots \\ R \end{bmatrix}_{Truth}$$

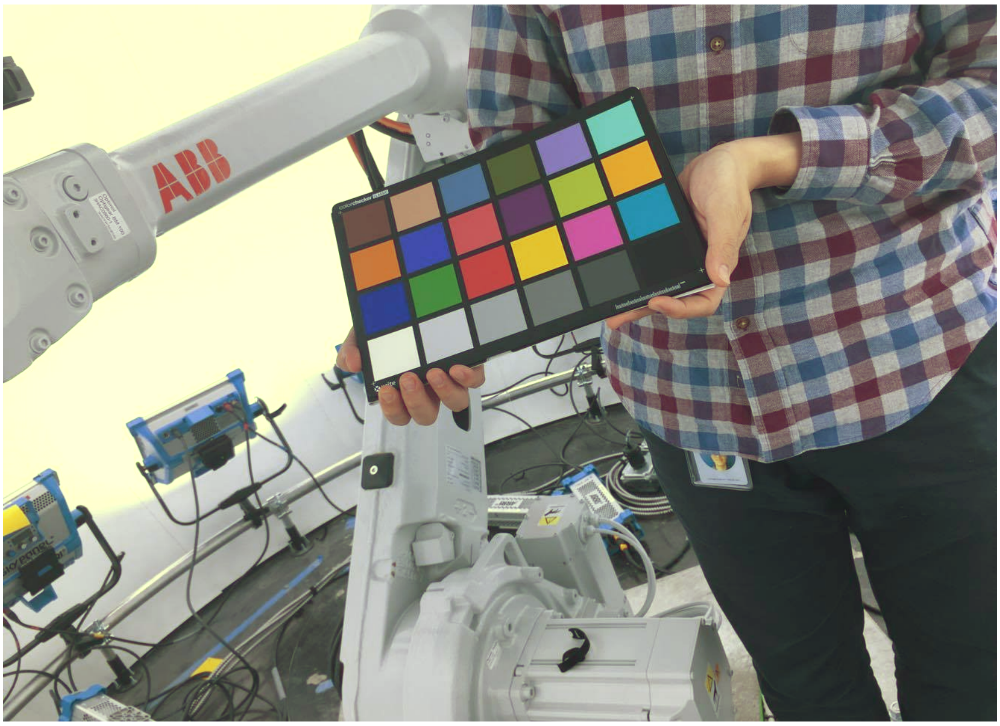
$$\begin{bmatrix} \alpha_6 \\ \vdots \\ 1 \end{bmatrix} = \begin{pmatrix} B^2 & G^2 & R^2 & B & G & C & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}_{Camera} / \begin{bmatrix} R \\ \vdots \\ R \end{bmatrix}_{Truth}$$

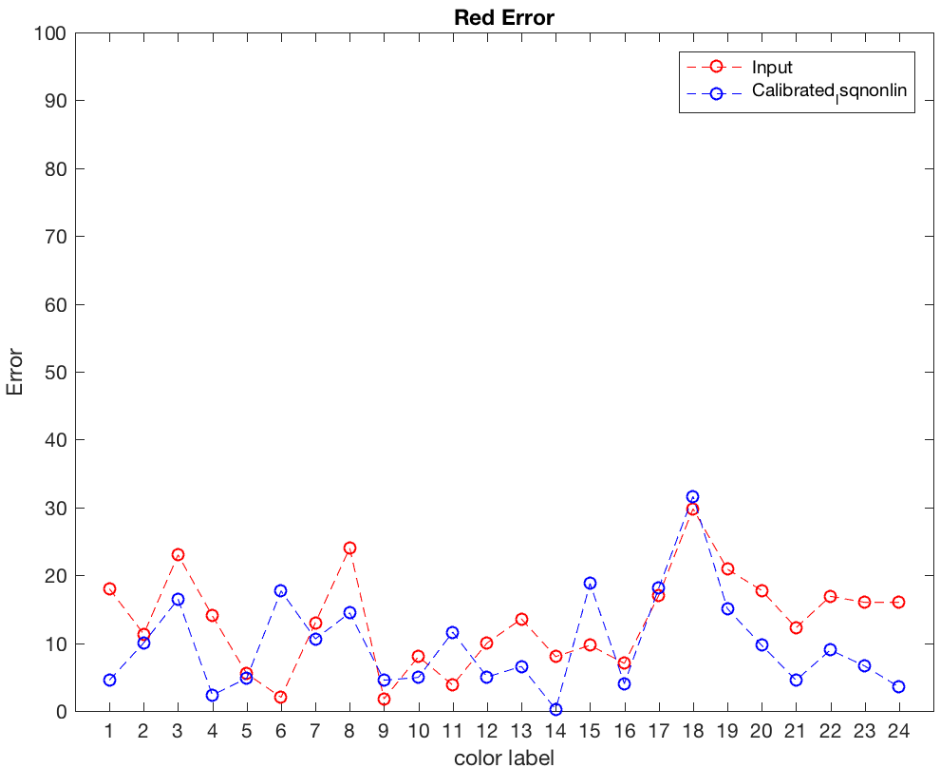
Experiments

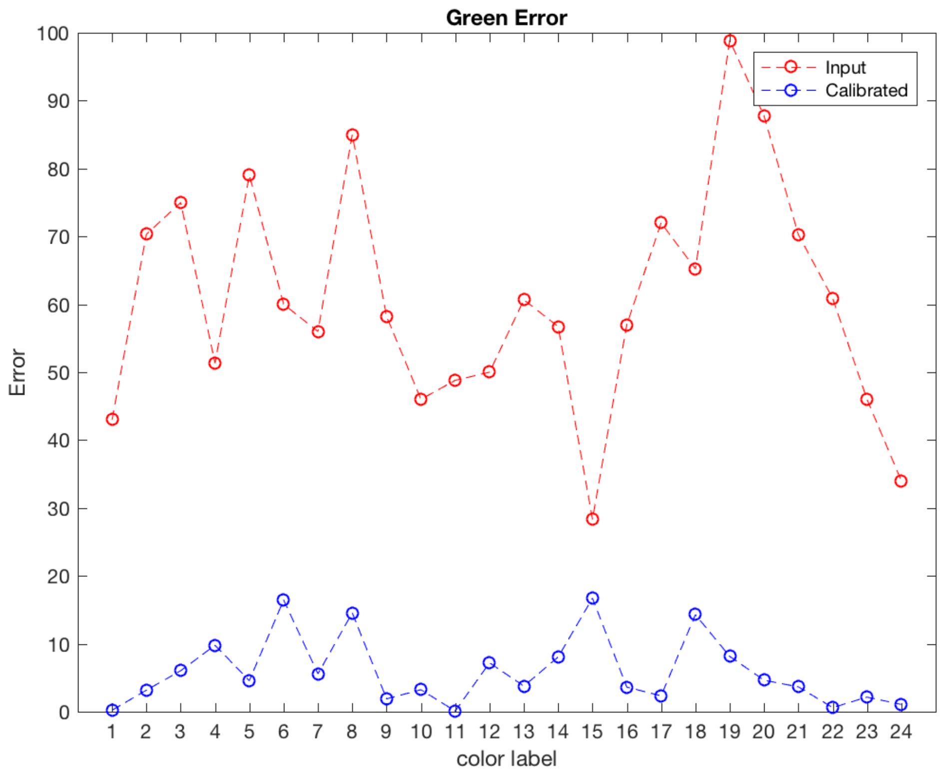
Input Image:

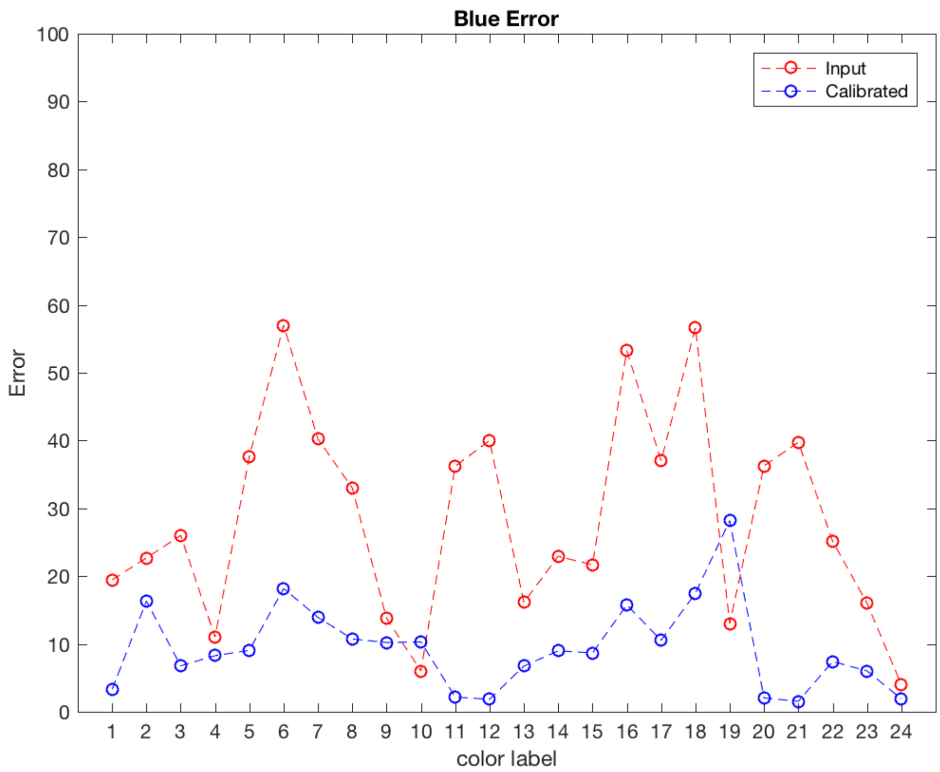


Calibrated Results:



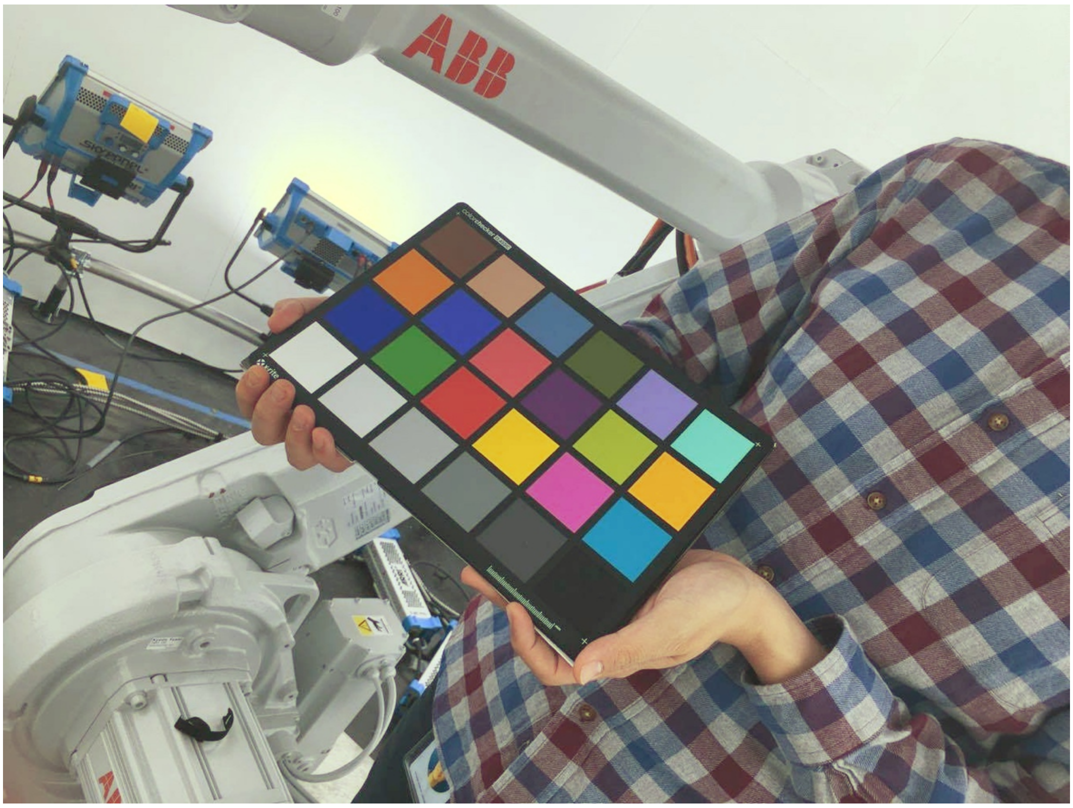


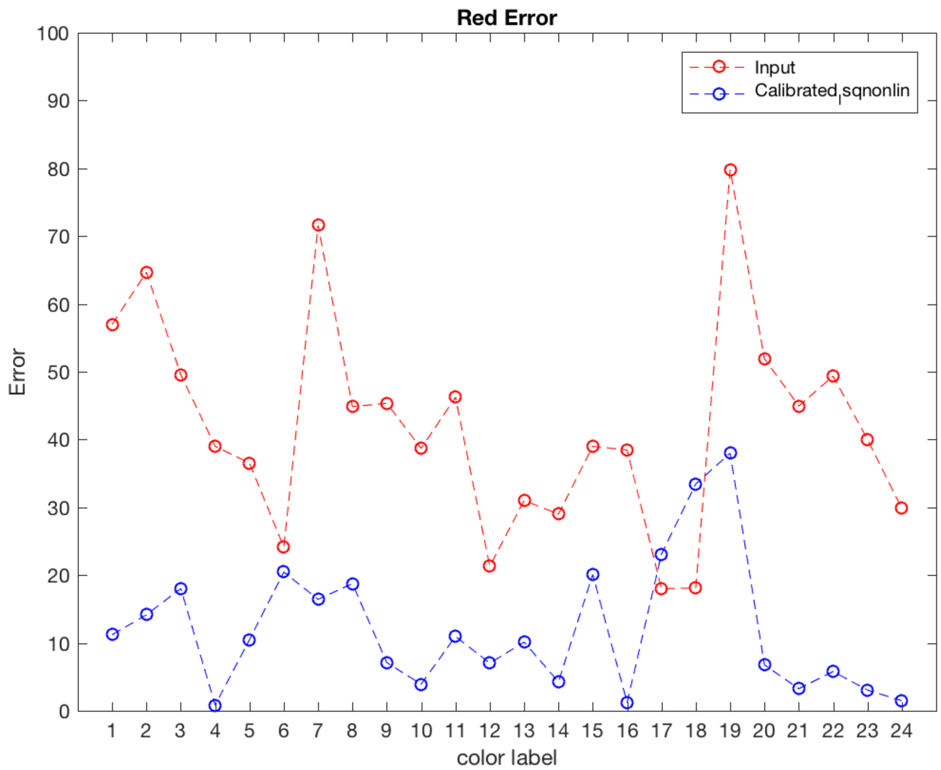

Green Error
Input
Calibrated
Error
color label
0
10
20
30
40
50
60
70
80
90
100
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

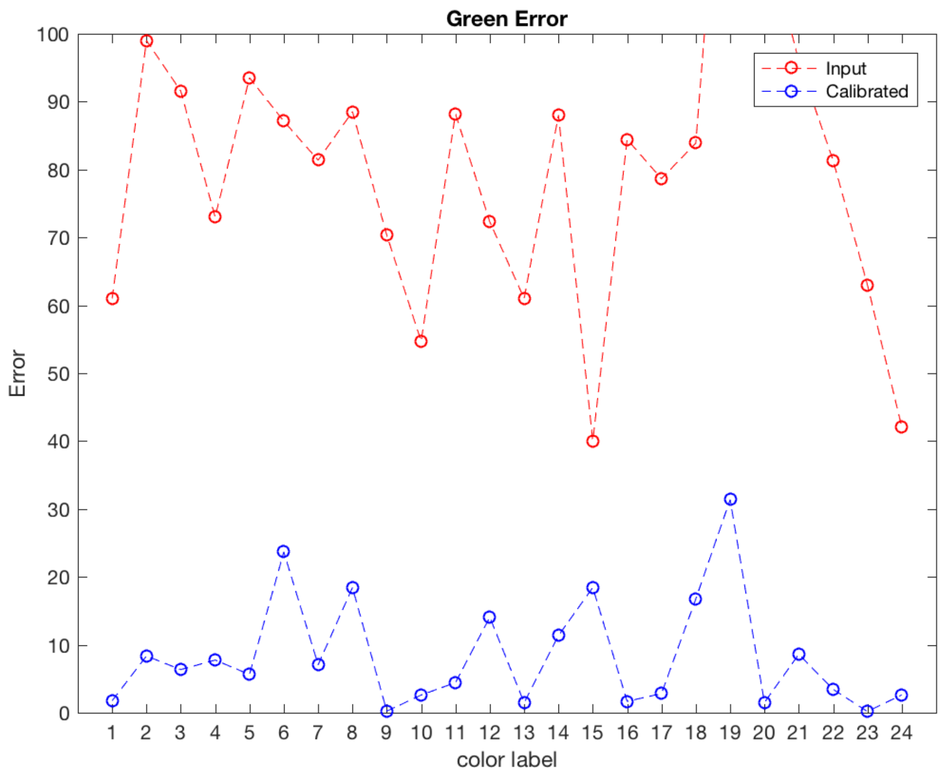

Blue Error
Input
Calibrated
Error
color label
0
10
20
30
40
50
60
70
80
90
100
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

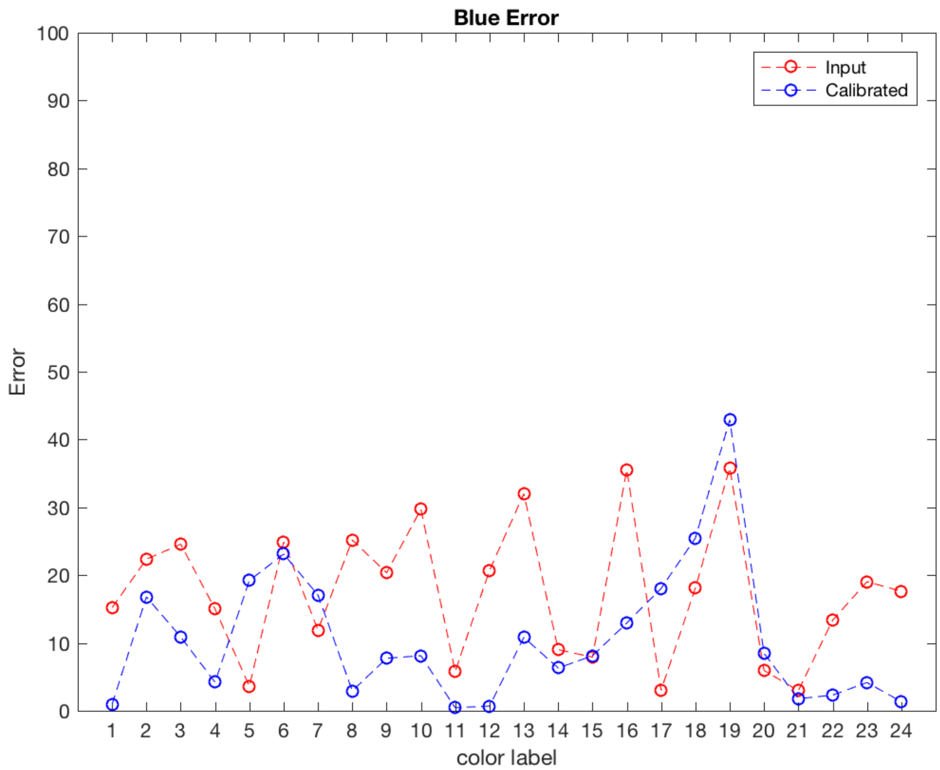
We can see that the result is very successful. The algorithm maps the purple light image back to an image that should look like one taken in illuminant D65 according to the Xrite document.

More results:




Red Error
Input
Calibrated|sqnonlin
Error
color label


Green Error
Input
Calibrated
Error
color label



We can also notice that since the purple light gives more errors/differences in green channel, the improvement in green channel is more apparent.

Mapping Function Optimization

On the other hand, the mapping function verification and optimization is still in progress. The current model are three multi-variable quadratic equations. However, I have been using linear method to solve the equations. To get better result, I try using non-linear method to refine the model of the linear solution.

$$argmin\|f(x;\theta) - y_i\|^2 = g(\theta)$$

$$\|f(x;\theta + \Delta\theta) - y_i\|^2 = \left\|f(x;\theta) + \frac{\partial f}{\partial \theta}\Delta\theta - y_i\right\|^2$$

$$= \|f(x;\theta) - yi + J(\theta)\Delta\theta\|^2, J(\theta) = \frac{\partial f}{\partial \theta}$$

$$\rightarrow \frac{\partial g}{\partial \theta} = 2J(\theta)^T\|f(x;\theta) - y + J(\theta)\Delta\theta\| = 0$$

$$\rightarrow 2J(\theta)^T f(x;\theta) - 2J(\theta)^T y + 2J(\theta)^T J(\theta)\Delta\theta = 0$$

$$\rightarrow \Delta\theta = \left(J(\theta)^T J(\theta)\right)^{-1}\left(J(\theta)^T y - J(\theta)^T f(x;\theta)\right)$$

$$= Hessian(\theta)^T J(\theta)^T (y - f(x;\theta))$$

I am trying to use the formula deducted above to get refine the solutions. I will compare the two results.

Challenge

1. In the last ILR:

   The matrix is not a square matrix and does not have an inverse matrix. Hence, I am using pseudo-inverse matrix to solve the problem. This can lead to some errors. I will try to use other polynomial fitting method to redo the algorithm and compare the results. → Currently in progress. This would be a challenge because it is hard to tell what model is better unless we try it. However, since there can be infinite number of models, it is hard to find the best one.

   Fitting Method:

$$\begin{pmatrix} B^2 & G^2 & R^2 & B & G & C & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}_{Camera} \begin{bmatrix} \alpha_6 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} R \\ \vdots \\ R \end{bmatrix}_{Truth}$$

$$\begin{bmatrix} \alpha_6 \\ \vdots \\ 1 \end{bmatrix} = \begin{pmatrix} B^2 & G^2 & R^2 & B & G & C & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}_{Camera} / \begin{bmatrix} R \\ \vdots \\ R \end{bmatrix}_{Truth}$$

Teamwork

Yiqing Cai	Generate one of the best path for robot arm. Test optimized path on the simplified dome setup.
Huan-Yang Chang	Simulation system setup, Robot arm manipulation. Test optimized path on the simplified dome setup.
Siddharth Raina	Sensor noise calibration plan development
Sambuddha Sarkar	Works on Blender. Generate virtual calibration target and render the virtual images of the target.

References

[1] https://en.wikipedia.org/wiki/Color_calibration

[2] https://en.wikipedia.org/wiki/Flood_fill