IRL #8: Progress Review

Man-Ning Chen (Mandy)

Team G: EXCALIBR

Teammates: Yiqing Cai, Huan-Yang Chang,

Siddharth Raina, Sambuddha Sarkar

Individual Progress

Overview

Since I have complete color checker detection and segmentation last week, I move on to searching for the color mapping function.

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.

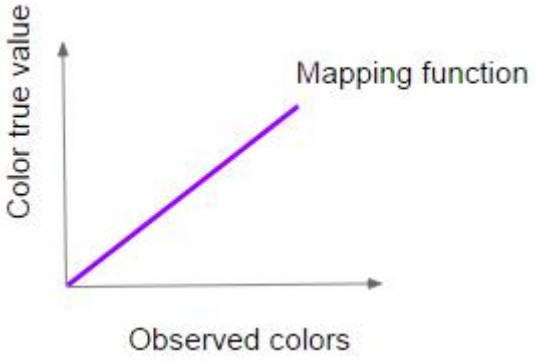


Figure 1. Mapping function (unreal): This graph is only for illustrating the concepts

CCFind

After Macduff algorithm failed in robust tests, I studied all the algorithms trying to build my own colorchecker detector. When I once again traced CCFind's code. I found that it did several downsampling in the code itself, which made me guess the reason it failed before is because of the image size. Therefore, I downsample my images first before applying CCFind on them. The results are good. Furthermore, since CCFind only use the contour information of the colorchecker, it would not be affected by bad lighting conditions.

Therefore, CCFind is the current algorithm used in my color calibration system. Figure 6 shows the algorithm.

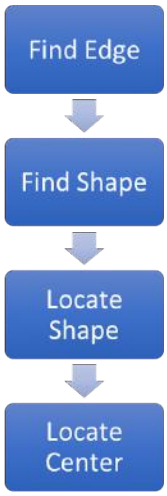
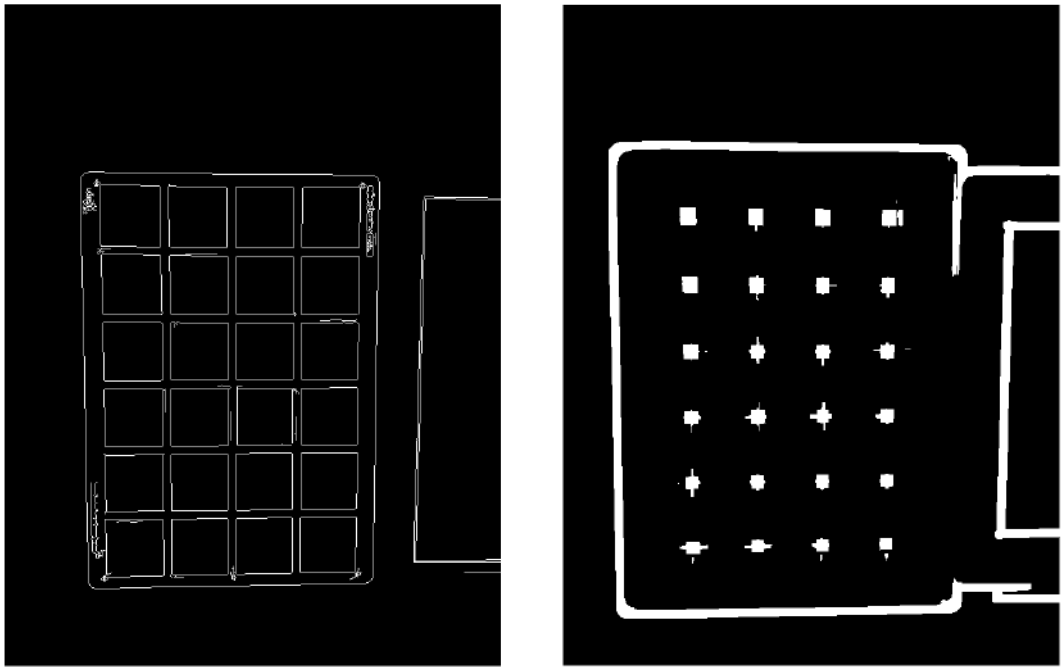


Figure 2. CCFind Algorithm



Find Edge Find Shape

Figure 3. How CCFind works

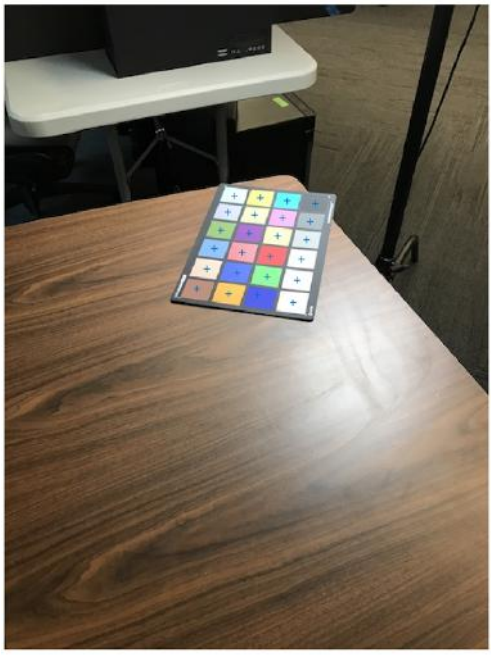


Image1

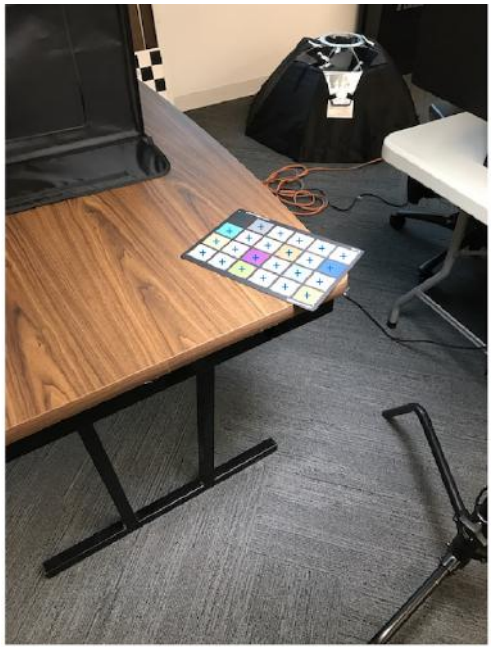


Image2

Figure 4. CCFind is robust while

1. the colorchecker only occupies small portion of the images
2. images are taken in special light conditions

Mapping function

First, I used different linear models to fit the data and calibrated the image with the mapping function model.

ColorChecker Information

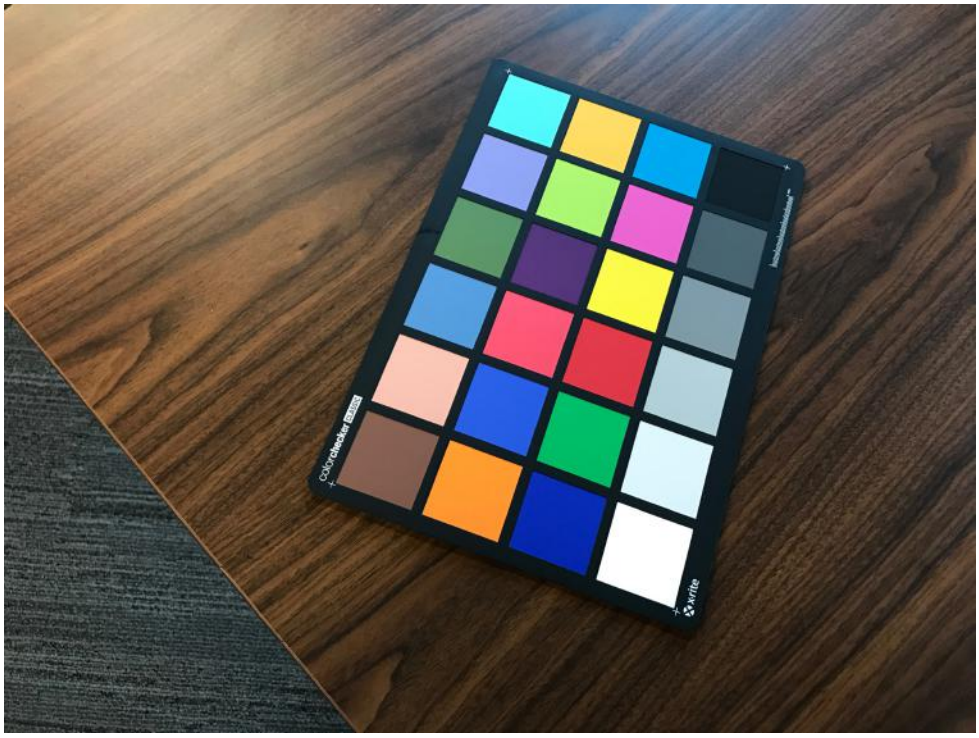
Figure 5. The 1-24 in this image correspond to the 1-24 in the x-axis below

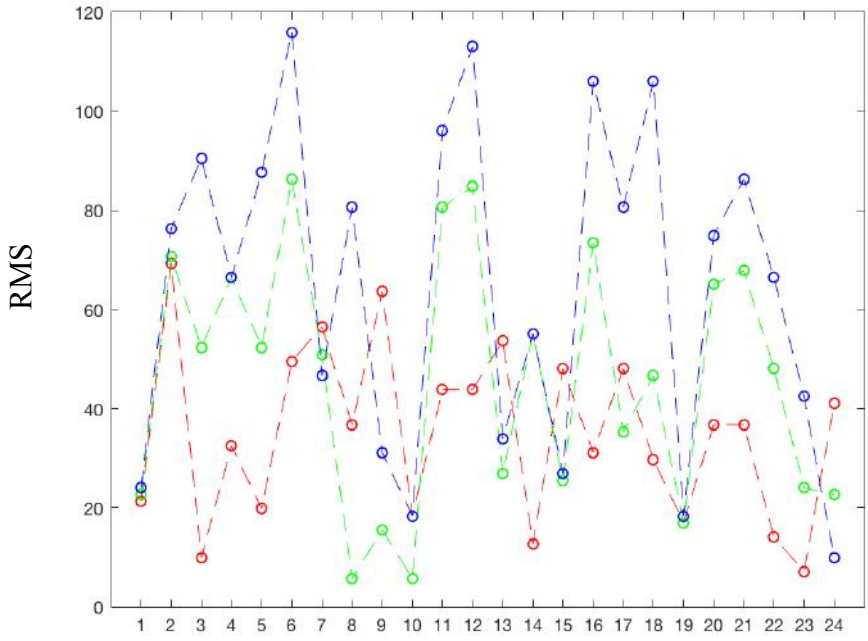
No.	Number	sRGB R	sRGB G	sRGB B	CIE L*a*b* L*	CIE L*a*b* a*	CIE L*a*b* b*	Munsell Notation Hue Value	Munsell Notation / Chroma
1.	dark skin	115	82	68	37.986	13.555	14.059	3 YR	3.7 / 3.2
2.	light skin	194	150	130	65.711	18.13	17.81	2.2 YR	6.47 / 4.1
3.	blue sky	98	122	157	49.927	-4.88	-21.925	4.3 PB	4.95 / 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 6. Color ground truth

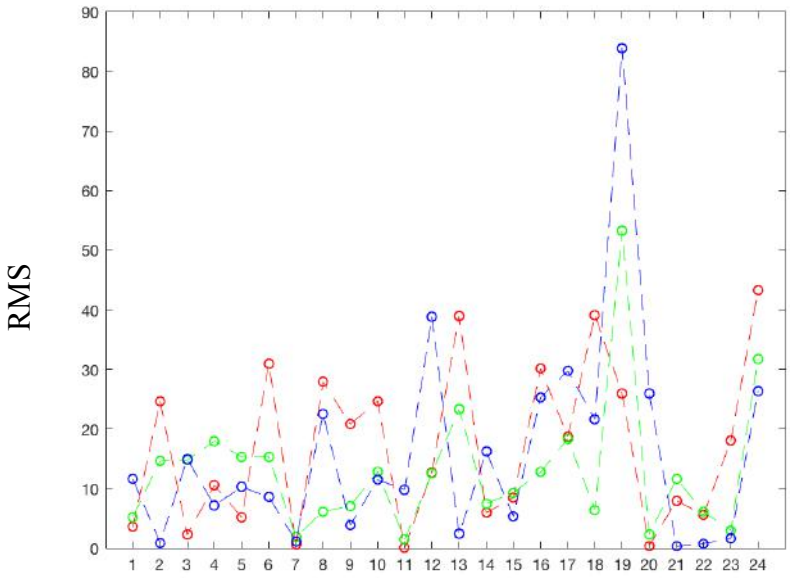
Input Image


ColorChecker
X-rite

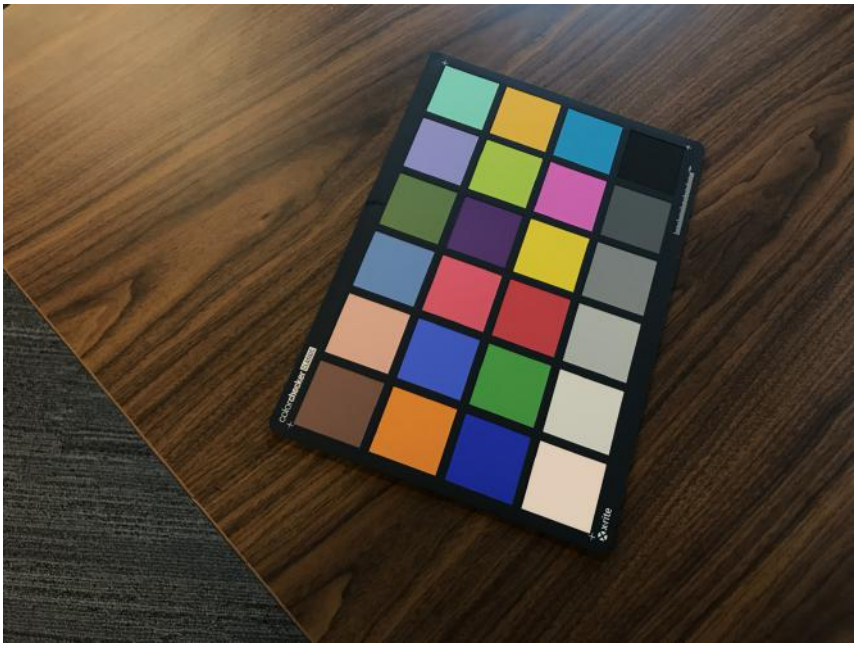

RMS
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Color Label

Model 1

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}_c = \begin{bmatrix} R \\ G \\ B \end{bmatrix}_T$$

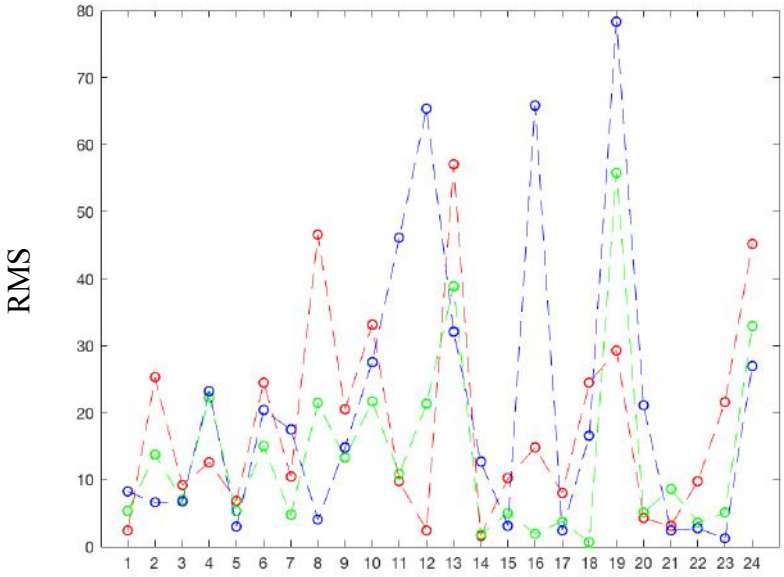


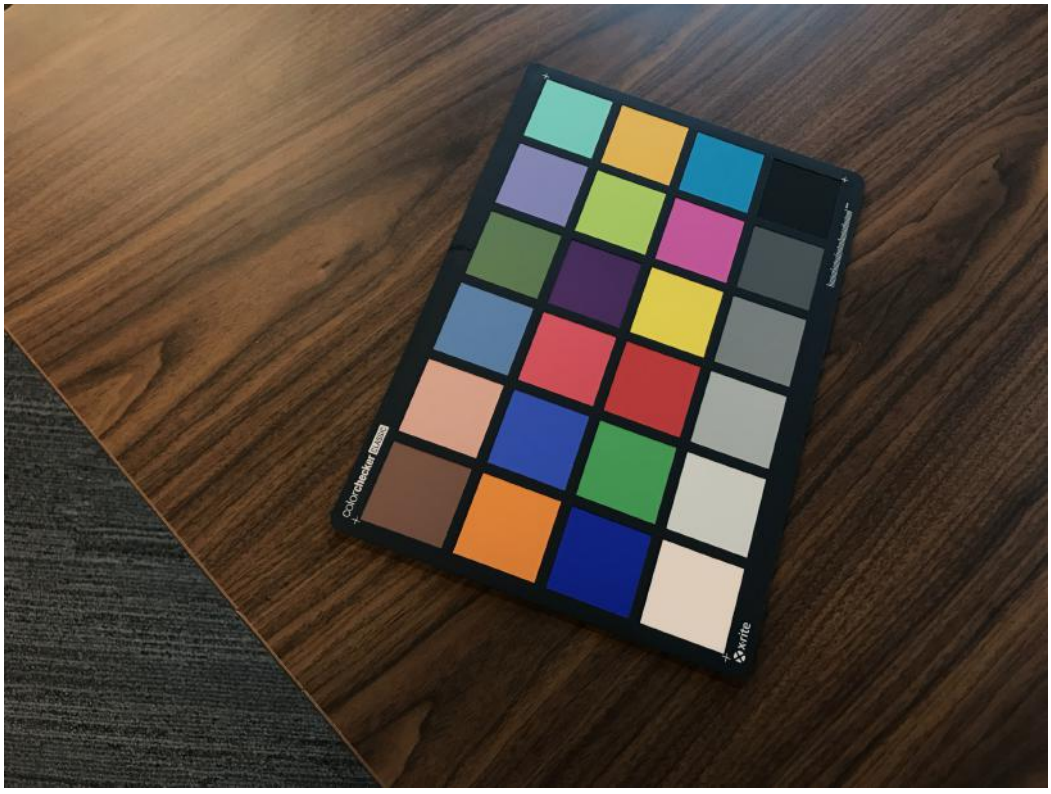
Calibrated image



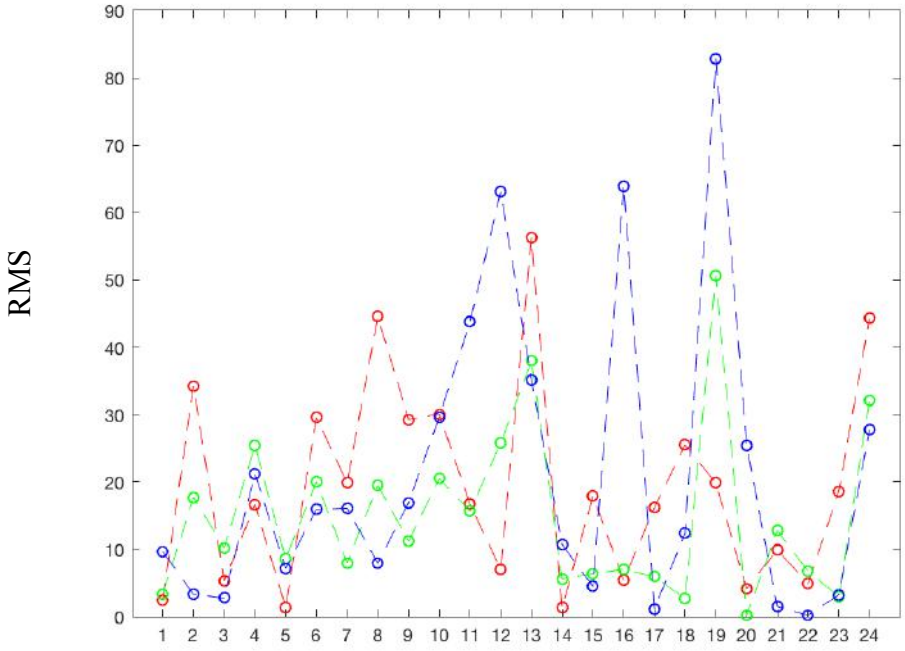
Model 2

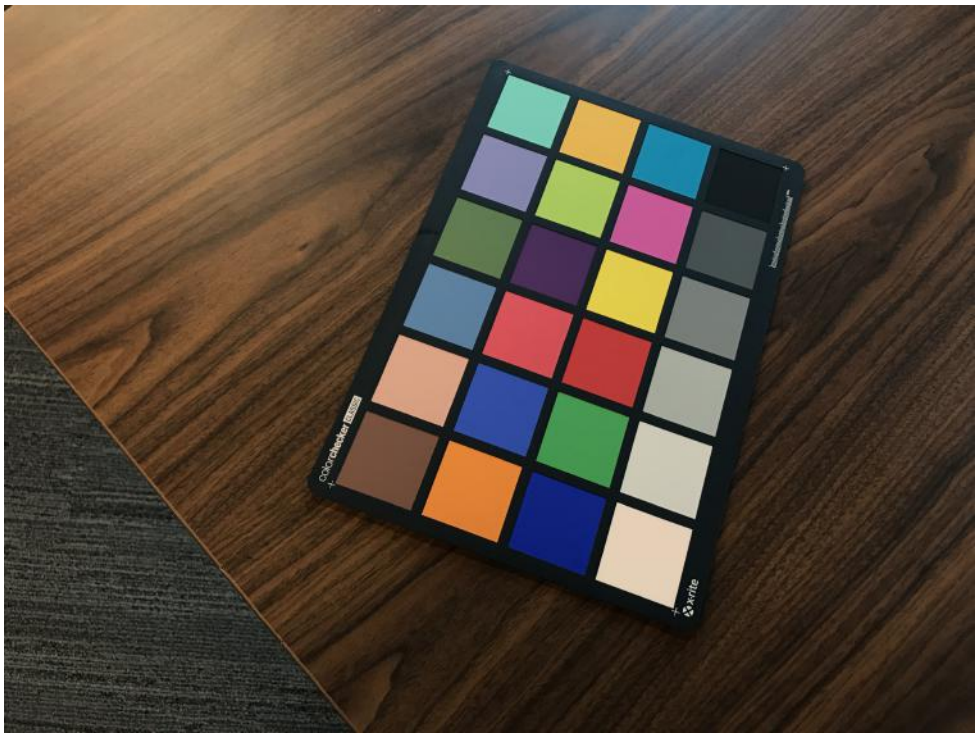
$$\begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}_c = \begin{bmatrix} R \\ G \\ B \end{bmatrix}_T$$





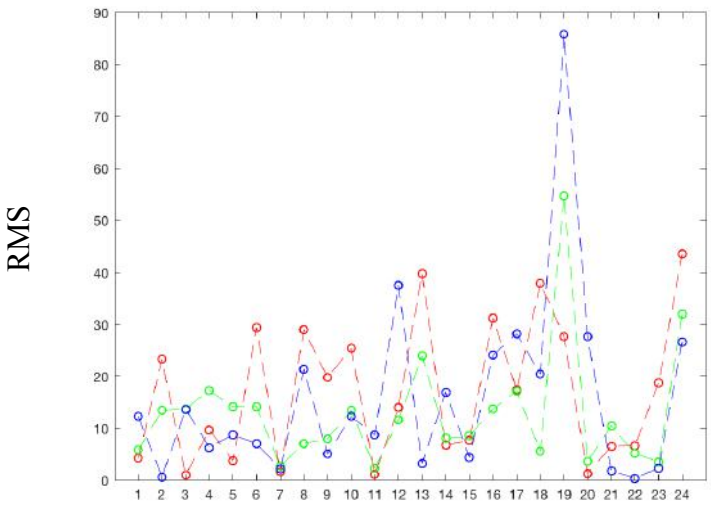
Using A\b in Model 2 would result in rank deficient warning. Therefore, I tried average camera observed RGB colors and Xrite ground truth values. Then, divide ground truth RGBs with observed RGBs. The result still looks incorrect.

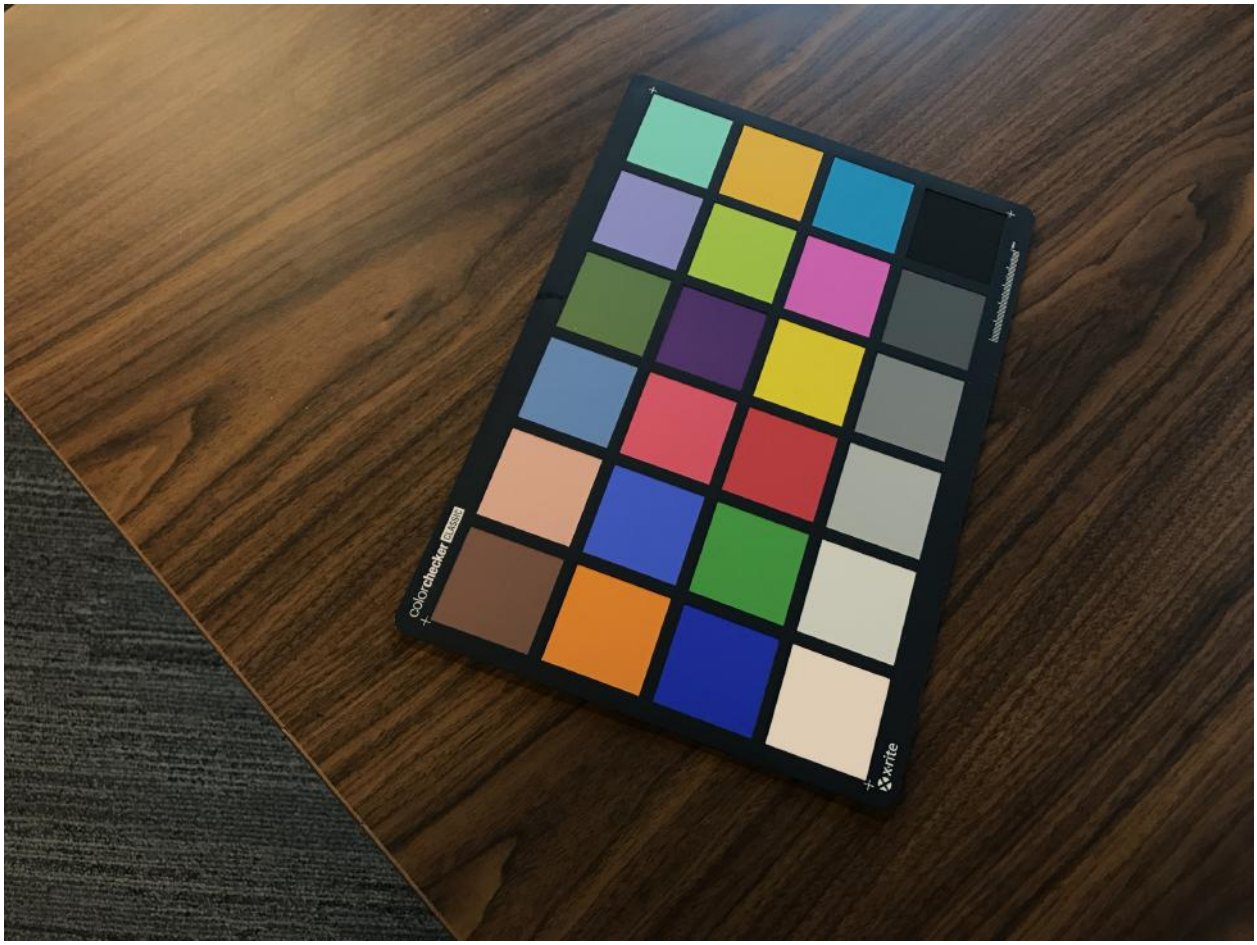

RMS
90
80
70
60
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Color Label



Model 3

$$\begin{bmatrix} 1+a_{11} & a_{12} & a_{13} \\ a_{21} & 1+a_{22} & a_{23} \\ a_{31} & a_{32} & 1+a_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}_c = \begin{bmatrix} R \\ G \\ B \end{bmatrix}_T$$





Challenge

1. In the robust tests conducted during this two weeks, the CCFind still fails in some test images with unknown reasons. I decide to find the mapping function first and then go back to modify CCFind.
2. A mapping model is not easy to find. I might need to do some researches in previous similar studies. However, it is not clear that if mapping models differ from camera to camera.

Teamwork

Yiqing Cai	Generate one of the best path for robot arm (go through as least positions as possible to cover FOVs of all cameras)
Huan-Yang Chang	Simulation system setup, Geometry calibration
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