Project 5 - Camera Characterization

Table of Contents

Molly Feldmann and Kevin Arnett, Team 13

step 3

```
img = imread('chart.png');
r = img(:,:,1);
g = img(:,:,2);
b = img(:,:,3);

row_start = [75 250 425 625];
row_end = [200 390 580 750];

col_start = [35 225 415 590 775 950];
col_end = [180 360 540 715 905 1075];

cam_rgbs = zeros(3,18);
for row = 1:3
    for col = 1:6
        cam_rgbs(1,((row*6)-6)+col) =
 mean2(r(row_start(row):row_end(row),col_start(col):col_end(col)))/255;
        cam_rgbs(2,((row*6)-6)+col) =
 mean2(g(row_start(row):row_end(row),col_start(col):col_end(col)))/255;
        cam_rgbs(3,((row*6)-6)+col) =
 mean2(b(row_start(row):row_end(row),col_start(col):col_end(col)))/255;
    end
end

cam_gray_rgbs = zeros(3,6);
for col = 1:6
```

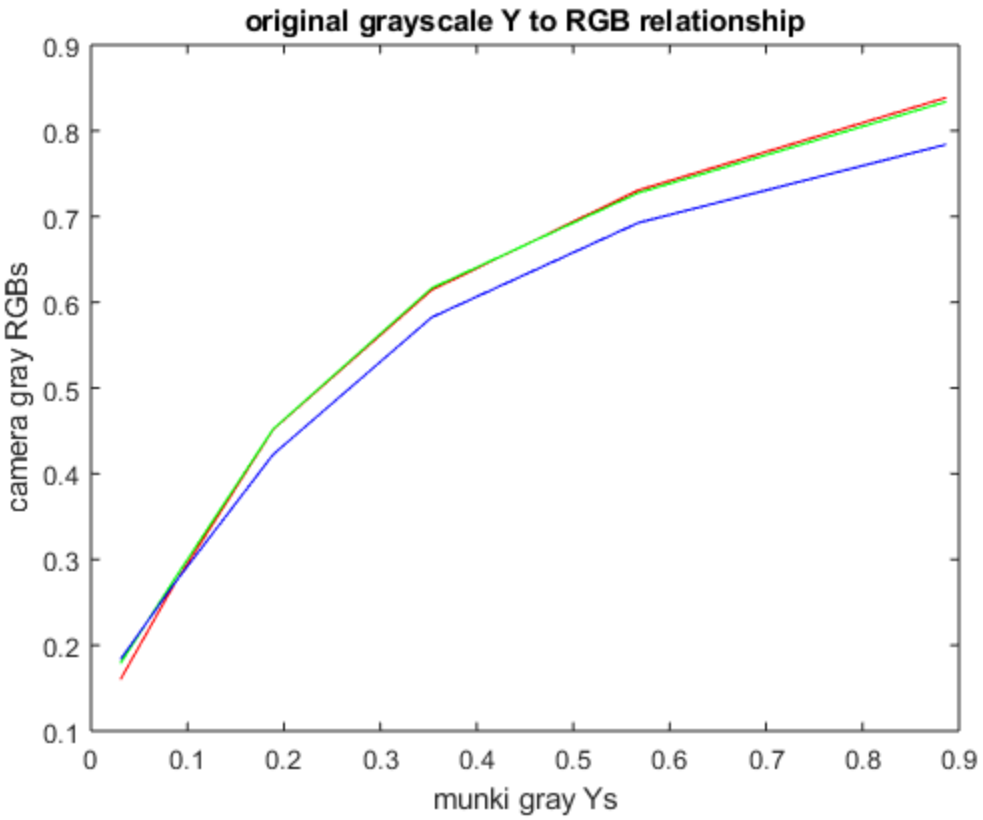
```
    cam_gray_rgbs(1,col) =
 mean2(r(row_start(4):row_end(4),col_start(col):col_end(col)))/255;
    cam_gray_rgbs(2,col) =
 mean2(g(row_start(4):row_end(4),col_start(col):col_end(col)))/255;
    cam_gray_rgbs(3,col) =
 mean2(b(row_start(4):row_end(4),col_start(col):col_end(col)))/255;
end

%cam_rgbs = [ cam_rgbs cam_gray_rgbs ];
cam_gray_rgbs = fliplr(cam_gray_rgbs);
```

step 4

```
munki = load("munki_CC_XYZs_Labs.txt");
munki_xyzs = munki(:,2:4)';
munki_labs = munki(:,5:7)';
munki_gray_ys = fliplr(munki_xyzs(2,19:24)/100);
```

step 5

```
figure;
plot(munki_gray_ys, cam_gray_rgbs(1,:), 'r',...
munki_gray_ys, cam_gray_rgbs(2,:), 'g',...
munki_gray_ys, cam_gray_rgbs(3,:), 'b');

title('original grayscale Y to RGB relationship');
xlabel('munki gray Ys');
ylabel('camera gray RGBs');
```

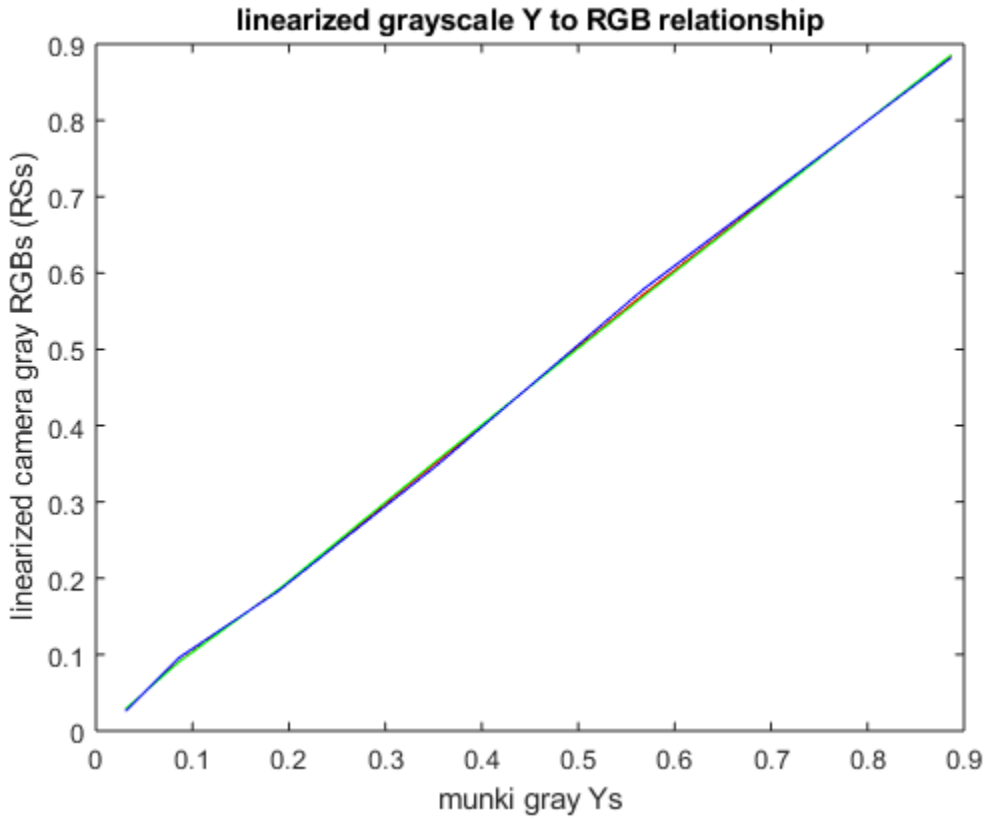


step 6

```
r = 1; g = 2; b = 3;

% a) fit low-order polynomial functions between normalized
% camera-captured gray RGBs and the munki-measured gray Ys
cam_polys(r,:) = polyfit(cam_gray_rgbs(r,:),munki_gray_ys,3);
cam_polys(g,:) = polyfit(cam_gray_rgbs(g,:),munki_gray_ys,3);
cam_polys(b,:) = polyfit(cam_gray_rgbs(b,:),munki_gray_ys,3);

% b) use the functions to linearize the camera data
cam_rss(r,:) = polyval(cam_polys(r,:),cam_rgbs(r,:));
cam_rss(g,:) = polyval(cam_polys(g,:),cam_rgbs(g,:));
cam_rss(b,:) = polyval(cam_polys(b,:),cam_rgbs(b,:));

cam_gray_rss(r,:) = polyval(cam_polys(r,:),cam_gray_rgbs(r,:));
cam_gray_rss(g,:) = polyval(cam_polys(g,:),cam_gray_rgbs(g,:));
cam_gray_rss(b,:) = polyval(cam_polys(b,:),cam_gray_rgbs(b,:));

% c) clip out of range values
cam_rss(cam_rss<0) = 0;
cam_rss(cam_rss>1) = 1;

cam_gray_rss(cam_gray_rss<0) = 0;
cam_gray_rss(cam_gray_rss>1) = 1;
```

step 7

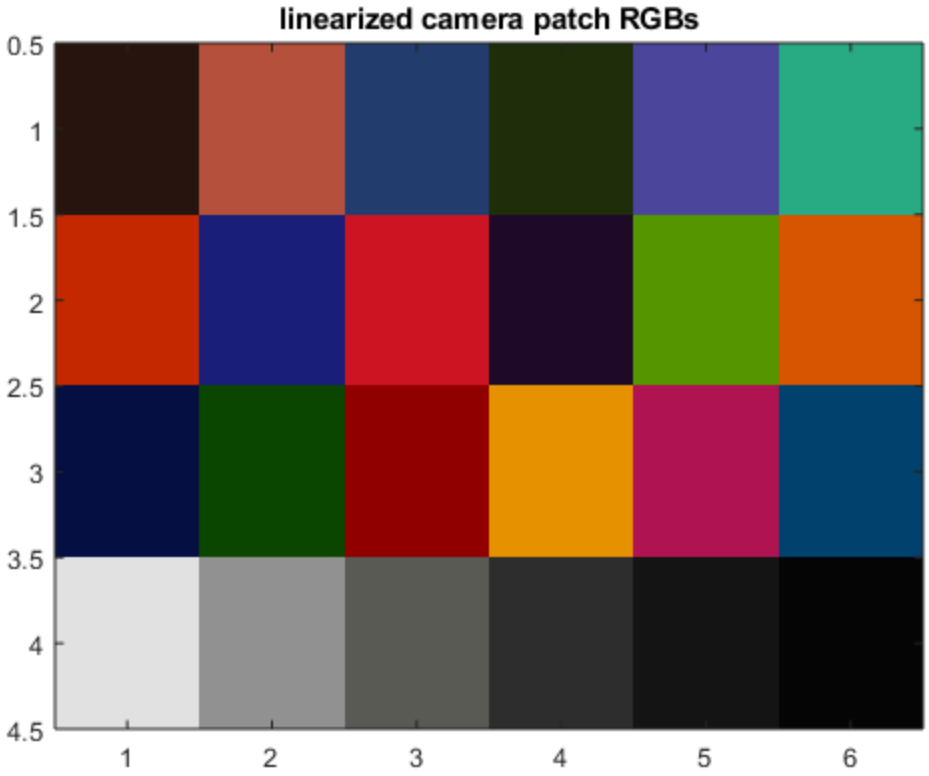
```
figure;
plot(munki_gray_ys, cam_gray_rss(1,:), 'r',...
munki_gray_ys, cam_gray_rss(2,:), 'g',...
munki_gray_ys, cam_gray_rss(3,:), 'b');

title('linearized grayscale Y to RGB relationship');
xlabel('munki gray Ys');
ylabel('linearized camera gray RGBs (RSs)');
```



step 8

```
% visualize the original camera RGBs
pix = permute([cam_rgbs fliplr(cam_gray_rgbs)], [3 2 1]);
pix = reshape(pix, [6 4 3]);
pix = imrotate(pix, -90);
pix = flipdim(pix,2);
figure;
image(pix);
title('original camera patch RGB');

% visualize the linearized camera RGBs
pix = permute([cam_rss fliplr(cam_gray_rss)], [3 2 1]);
pix = reshape(pix, [6 4 3]);
```

```
pix = imrotate(pix, -90);
pix = flipdim(pix,2);
figure;
image(pix);
title('linearized camera patch RGBs');
```



step 9

```
% use the munki-measured ColorChecker XYZs and camera-captured RGB RSs
 to
% derive a 3x3 matrix that can be used to estimate XYZs from camera
 RGBs
cam_matrix3x3 = munki_xyzs * pinv([cam_rss fliplr(cam_gray_rss)]);

disp(cam_matrix3x3);
```

```
   39.1635   41.3495    7.4459
   23.0005   66.9933    1.8758
    3.2714   13.5379   59.2652
```

step 10

```
% estimate the ColorChecker XYZs from the linearized camera rgbs using
% the 3x3 camera matrix
cam_xyzs = cam_matrix3x3 * [cam_rss fliplr(cam_gray_rss)];

disp(cam_xyzs);
```

```
  Columns 1 through 7

   10.9415   43.2150   19.1768   12.5306   27.8833   38.8629   37.1904
```

```
   10.1661   38.3927   20.4907   14.8900   26.6280   50.7113   28.9003
    5.2507   20.9408   29.6720    5.6610   41.3326   40.4021    4.7732

  Columns 8 through 14

   12.4482   36.3302    8.5356   38.1840   48.1319    5.0933   13.6300
   11.2658   24.7887    6.7322   47.9058   43.1702    4.8098   19.9290
   30.7791   12.0891   10.7038    9.2057    7.5391   16.4569    4.1416

  Columns 15 through 21

   22.8687   59.6334   33.2577   14.5582   77.8516   50.2272   31.0255
   13.4141   59.9585   22.7413   18.4748   81.3559   52.3466   32.4901
    2.3004   10.8689   22.2653   29.3411   67.1613   43.8560   26.5564

  Columns 22 through 24

   16.2639    7.9692    2.5104
   16.9964    8.2784    2.6432
   13.9626    7.1733    2.0277
```

step 11

```
% use XYZ2Lab function to calculate Lab values from the estimated XYZ
% values
cie.cmf2deg = load("CIE_2Deg_380-780-5nm.txt");
cie.cmf2deg = cie.cmf2deg(:,2:4);
cie.illD50 = load("CIE_IllD50_380-780-5nm.txt");
cie.illD50 = cie.illD50(:,2);
cie.illE = ones(81,1);

XYZn = ref2XYZ(cie.illE,cie.cmf2deg,cie.illD50);
cam_labs = XYZ2Lab(cam_xyzs,XYZn);

% calculate DeltaEab color differences between these estimated Lab
 values
% and the measured Lab values in the “munki_Labs” variable
delta_eab = deltaEab(cam_labs, munki_labs);

% call print_camera_model_table function supplied in the resources to
 print
% a table
print_camera_model_table(munki_labs, cam_labs, delta_eab);
```

```
Camera model color error
camera->camera_RGBs->camera_model->estimated_XYZs

 colormunki measured vs. camera estimated ColorChecker Lab values
     measured       estimated
patch #      L        a        b        L        a        b       dEab
```

```
     1   37.1865  14.9985  15.2592  38.1388   8.7132  13.5002
6.5959
     2   65.8188  16.8695  18.0267  68.3091  19.2422  18.7430
3.5135
     3   49.9949  -3.1841 -23.5159  52.3875  -2.9139 -24.3053
2.5339
     4   42.6411 -15.3251  20.0423  45.4832 -11.7498  24.1350
6.1327
     5   54.6852   9.6978 -26.7126  58.6284   8.9727 -30.1583
5.2866
     6   71.2441 -33.1391  -0.5010  76.5038 -29.3819   1.8622
6.8823
     7   62.2558  34.1094  57.7774  60.6935  33.3894  54.8854
3.3650
     8   39.5890   9.9980 -43.6388  40.0246  11.2220 -47.3679
3.9490
     9   51.8424  48.1403  16.0636  56.8690  47.0466  20.2071
6.6055
    10   29.4495  22.4255 -21.7661  31.1893  19.4378 -19.8766
3.9400
    11   71.6264 -24.3441  57.6850  74.7655 -24.0525  60.2169
4.0434
    12   72.2288  20.6039  69.0149  71.6703  18.7480  61.0809
8.1673
    13   28.6402  18.5907 -51.4092  26.1858   5.7716 -44.1112
14.9537
    14   54.6309 -39.5493  32.8341  51.7568 -31.5909  43.0511
13.2658
    15   42.5988  54.6049  25.7315  43.3807  53.5508  41.7396
16.0617
    16   82.4265   3.8689  78.8570  81.8156   4.3838  66.8923
11.9914
    17   51.5476  49.5154 -14.3758  54.8049  45.4618  -7.1552
8.8982
    18   49.3892 -26.5473 -28.6645  50.0670 -18.5241 -27.7756
8.1007
    19   95.4458  -0.4414   0.0244  92.2898  -1.1748  -0.0184
3.2403
    20   80.0339   0.1309  -0.9345  77.4877  -0.6526  -0.8109
2.6669
    21   66.0107  -0.0004  -1.1463  63.7459  -1.1049   0.4409
2.9779
    22   50.5546  -0.6207  -0.9616  48.2556  -0.6997   0.1694
2.5635
    23   35.1532  -0.0632  -0.9708  34.5561  -0.1161  -1.4280
0.7539
    24   20.3224  -0.2858  -0.5603  18.5543  -0.7474   1.4336
2.7046

      min      0.7539
      max     16.0617
      mean     6.2164
```

step 12

```
% split the radiometric scalars (cam_RSs) into r,g,b vectors
rsrgbs = [cam_rss fliplr(cam_gray_rss)];
rsrs = rsrgbs(1,:);
rsgs = rsrgbs(2,:);
rsbs = rsrgbs(3,:);

% create vectors of these RSs with multiplicative terms to
% represent interactions and square terms to represent non-linearities
 in
% the RGB-to-XYZ relationship
rsrgbs_extd = [rsrgbs; rsrs.*rsgs; rsrs.*rsbs; rsgs.*rsbs;
 rsrs.*rsgs.*rsbs; ...
rsrs.^2; rsgs.^2; rsbs.^2; ones(1,size(rsrgbs,2))];

% find the extended (3x11) matrix that relates the RS and XYZ datasets
cam_matrix3x11 = munki_xyzs * pinv(rsrgbs_extd);

disp(cam_matrix3x11);
```

```
  Columns 1 through 7

   47.9944   41.6204   11.0316   32.2842  -28.2214   11.0860   46.5505
   25.7529   74.7982    3.2524   25.9622  -22.2312    9.3286   46.8712
    4.9270   19.2506   79.0055   -7.4882  -14.0103  -14.8008   78.6477

  Columns 8 through 11

  -16.6072  -31.5183  -12.2403   -0.1408
   -9.0883  -38.0562  -10.6880   -0.0937
   -2.5317   -5.8184  -46.8160   -0.3287
```

step 13

```
cam_extd_xyzs = cam_matrix3x11 * rsrgbs_extd;

disp(cam_extd_xyzs);
```

```
  Columns 1 through 7

   12.0840   43.7924   18.6750   13.0826   25.8270   31.8609   37.2363
   10.9917   39.3088   20.0423   15.7805   24.5858   43.4868   29.0559
    6.3307   22.8211   28.7536    6.8810   35.0327   36.6304    4.0326

  Columns 8 through 14

   11.3929   32.2838    9.2189   34.7025   49.2428    5.1956   12.0139
    9.9530   21.3238    6.8963   44.1354   44.6002    4.7844   19.5562
   28.5493   12.2407   12.4008    8.9424    5.7966   18.1406    4.9752
```

```
  Columns 15 through 21

   22.1491   60.3524   28.1658   13.0423   85.7948   51.2134   31.7977
   11.7611   60.2161   18.0411   17.5665   89.2245   53.1820   33.1119
    2.1518    7.3513   21.5454   28.2007   73.8069   43.6252   27.9328

  Columns 22 through 24

   17.2227    8.6420    2.6905
   17.8814    8.9226    2.8561
   16.0167    8.6504    2.3606
```

step 14

```
% use the XYZ2Lab function to calculate Lab values from the estimated
 XYZ
% values
cam_extd_labs = XYZ2Lab(cam_extd_xyzs,XYZn);

% calculate DeltaE color differences between these estimated Lab
 values and
% the measured Lab values provided in the file
 “munki_CC_XYZs_Labs.txt”
extd_delta_eab = deltaEab(cam_extd_labs, munki_labs);

% use the print_extended_camera_model_table function provided in the
% resources to print a table like the one shown below. The min, max,
 and
% mean differences should all be smaller than the ones you calculated
 in
% step 11
print_extended_camera_model_table(munki_labs, cam_extd_labs,
 extd_delta_eab);
```

```
Extended camera model color error
camera->camera_RGBs->extended_camera_model->estimated_XYZs

 colormunki measured vs. camera estimated ColorChecker Lab values
     measured       estimated
patch #      L        a        b        L        a        b       dEab
      1   37.1865  14.9985  15.2592  39.5665  10.7080  10.8248
 6.6134
      2   65.8188  16.8695  18.0267  68.9744  18.0711  16.2084
 3.8351
      3   49.9949  -3.1841 -23.5159  51.8851  -3.3162 -23.6890
 1.9027
      4   42.6411 -15.3251  20.0423  46.6852 -13.2653  20.7042
 4.5864
      5   54.6852   9.6978 -26.7126  56.6696   9.0791 -25.0174
 2.6823
```

```
     6   71.2441 -33.1391  -0.5010  71.8841 -33.1355  -1.0365
0.8345
     7   62.2558  34.1094  57.7774  60.8310  32.9469  59.3492
2.4191
     8   39.5890   9.9980 -43.6388  37.7580  13.6394 -47.7119
5.7622
     9   51.8424  48.1403  16.0636  53.3020  48.4826  13.6183
2.8683
    10   29.4495  22.4255 -21.7661  31.5698  23.5925 -24.3108
3.5118
    11   71.6264 -24.3441  57.6850  72.3189 -25.0238  56.9252
1.2324
    12   72.2288  20.6039  69.0149  72.6279  17.6492  70.2879
3.2419
    13   28.6402  18.5907 -51.4092  26.1116   7.3390 -48.0954
11.9990
    14   54.6309 -39.5493  32.8341  51.3316 -40.4890  37.6679
5.9273
    15   42.5988  54.6049  25.7315  40.8339  61.2474  38.6837
14.6628
    16   82.4265   3.8689  78.8570  81.9555   5.4863  79.5684
1.8286
    17   51.5476  49.5154 -14.3758  49.5460  49.2337 -14.8142
2.0683
    18   49.3892 -26.5473 -28.6645  48.9660 -23.3604 -27.8138
3.3255
    19   95.4458  -0.4414   0.0244  95.6742  -0.4378  -0.1495
0.2871
    20   80.0339   0.1309  -0.9345  77.9823  -0.1689   0.3267
2.4268
    21   66.0107  -0.0004  -1.1463  64.2514  -0.4643  -1.0157
1.8241
    22   50.5546  -0.6207  -0.9616  49.3520  -0.1002  -3.1185
2.5237
    23   35.1532  -0.0632  -0.9708  35.8348   0.3361  -4.9289
4.0362
    24   20.3224  -0.2858  -0.5603  19.4583  -1.1813  -0.0299
1.3528

      min      0.2871
      max     14.6628
      mean     3.8230
```

step 15

```
% save the (extended) camera model for use in later projects
save('cam_model.mat', 'cam_polys', 'cam_matrix3x11');
```

step 16

```
% <include>camRGB2XYZ.m</include>
```

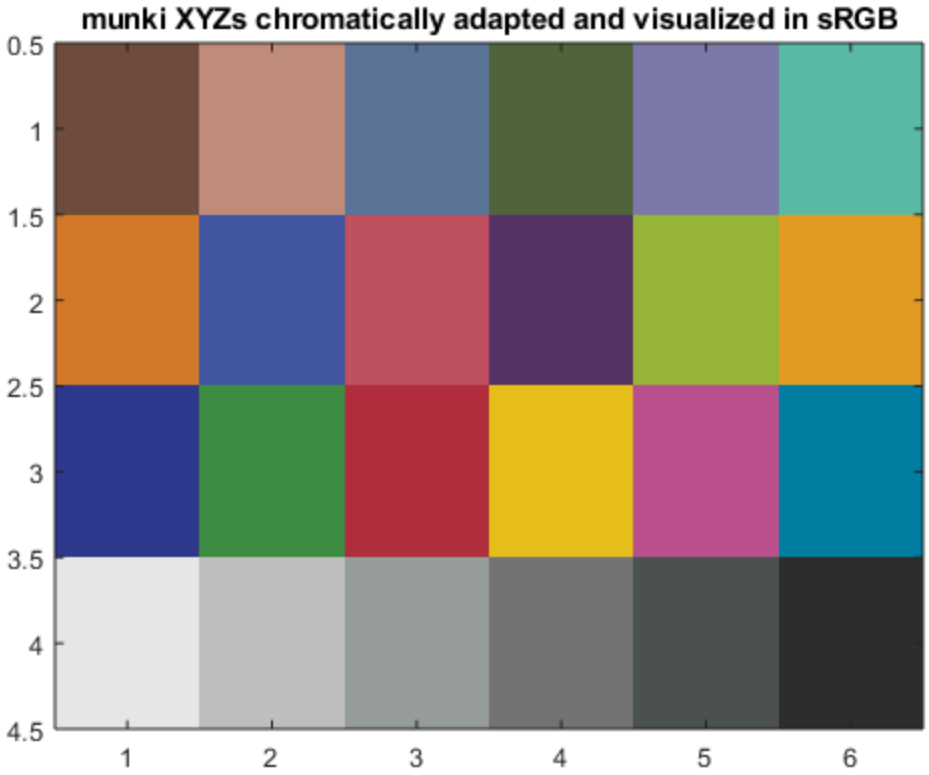
```
cam_XYZs = camRGB2XYZ('cam_model.mat', [cam_rgbs
 fliplr(cam_gray_rgbs)]);

disp(cam_XYZs);
```

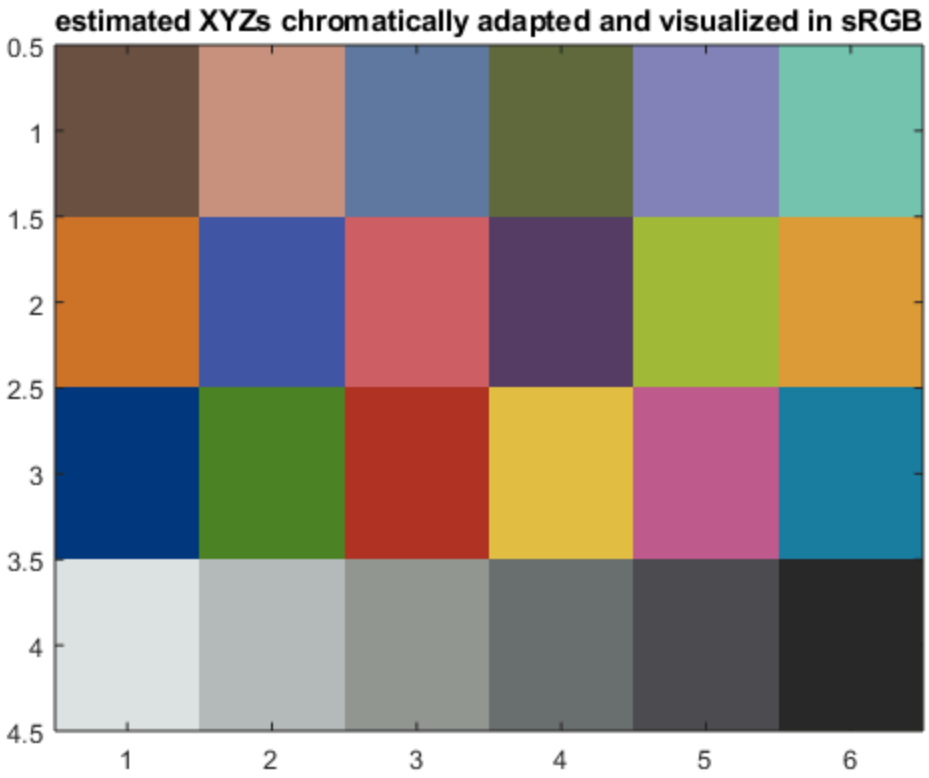
```
  Columns 1 through 7

   12.0840   43.7924   18.6750   13.0826   25.8270   31.8609   37.2363
   10.9917   39.3088   20.0423   15.7805   24.5858   43.4868   29.0559
    6.3307   22.8211   28.7536    6.8810   35.0327   36.6304    4.0326

  Columns 8 through 14

   11.3929   32.2838    9.2189   34.7025   49.2428    5.1956   12.0139
    9.9530   21.3238    6.8963   44.1354   44.6002    4.7844   19.5562
   28.5493   12.2407   12.4008    8.9424    5.7966   18.1406    4.9752

  Columns 15 through 21

   22.1491   60.3524   28.1658   13.0423   85.7948   51.2134   31.7977
   11.7611   60.2161   18.0411   17.5665   89.2245   53.1820   33.1119
    2.1518    7.3513   21.5454   28.2007   73.8069   43.6252   27.9328

  Columns 22 through 24

   17.2227    8.6420    2.6905
   17.8814    8.9226    2.8561
   16.0167    8.6504    2.3606
```

step 17

```
XYZ_D50 = XYZn;

cie.illD65 = load("CIE_IllD65_380-780-5nm.txt");
cie.illD65 = cie.illD65(:,2);
XYZ_D65 = ref2XYZ(cie.illE,cie.cmf2deg,cie.illD65);

% visualize the munki-measured XYZs as an sRGB image
munki_XYZs_D65 = catBradford(munki_xyzs, XYZ_D50, XYZ_D65);
munki_XYZs_sRGBs = XYZ2sRGB(munki_XYZs_D65);
pix = reshape(munki_XYZs_sRGBs', [6 4 3]);
pix = uint8(pix*255);
pix = imrotate(pix, -90);
pix = flipdim(pix,2);
figure;
image(pix);
title('munki XYZs chromatically adapted and visualized in sRGB');

% visualize the camera-estimated XYZs as an sRGB image
cam_XYZs_D65 = catBradford(cam_xyzs, XYZ_D50, XYZ_D65);
cam_XYZs_sRGBs = XYZ2sRGB(cam_XYZs_D65);
pix = reshape(cam_XYZs_sRGBs', [6 4 3]);
```

```
pix = uint8(pix*255);
pix = imrotate(pix, -90);
pix = flipdim(pix,2);
figure;
image(pix);
title('estimated XYZs chromatically adapted and visualized in sRGB');
```





step 18

who did what parts of the project:

- step 3 - Kevin
- step 4 - Kevin
- step 5 - Kevin
- step 6 - Kevin
- step 7 - Kevin
- step 8 - Kevin
- step 9 - Kevin
- step 10 - Kevin
- step 11 - Molly
- step 12 - Kevin
- step 13 - Kevin
- step 14 - Molly

- step 15 - Kevin
- step 16 - Molly
- step 17 - Molly
- step 18 - Kevin and Molly

any problems you had with the project:

any parts of the project you thought were valuable:

any improvements you'd like to see:

Published with MATLAB® R2018a