Project 3 Report

Table of Contents

Authors: Molly Feldmann and Kevin Arnett

Team: 13

Step 2: LoadCIEData - Function

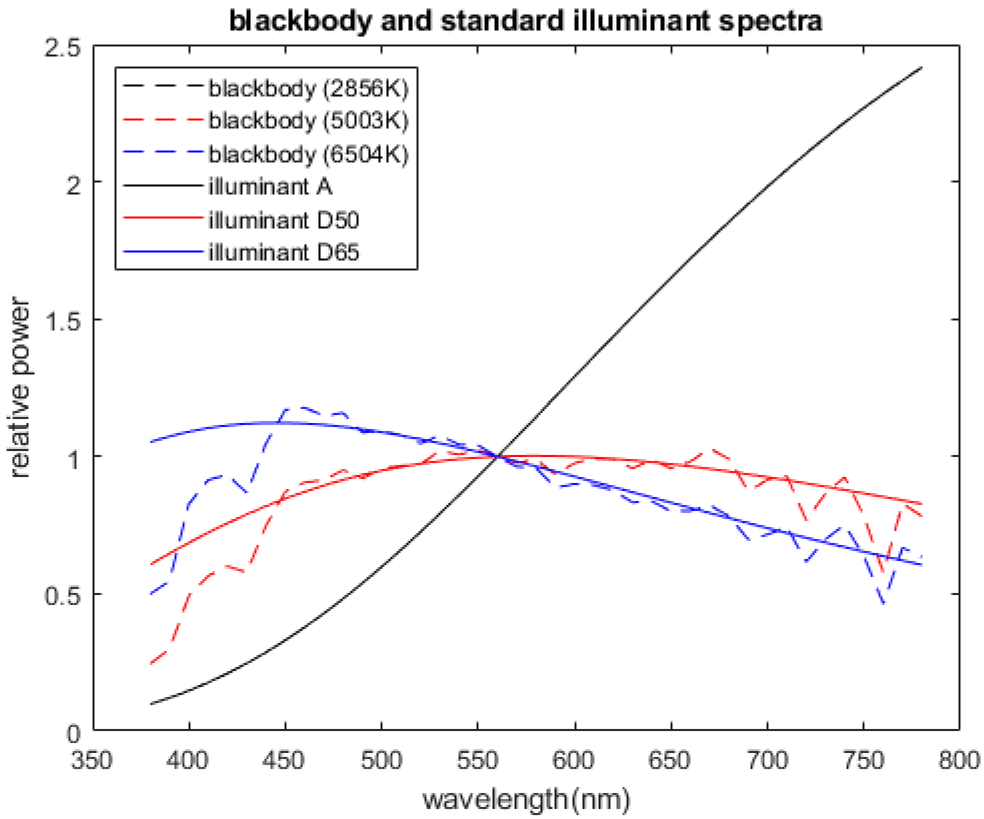
```
function [cie] = loadCIEdata()
    cie.lambda = load("CIE_2Deg_380-780-5nm.txt");
    cie.lambda = cie.lambda(:,1)';
    cie.cmf2deg = load("CIE_2Deg_380-780-5nm.txt");
    cie.cmf2deg = cie.cmf2deg(:,2:4);
    cie.cmf10deg = load("CIE_10Deg_380-780-5nm.txt");
    cie.cmf10deg = cie.cmf10deg(:,2:4);
    cie.illA = load("CIE_IllA_380-780-5nm.txt");
    cie.illA = cie.illA(:,2);
    cie.illC = load("CIE_IllC_380-780-5nm.txt");
    cie.illC = cie.illC(:,2);
    cie.illD50 = load("CIE_IllD50_380-780-5nm.txt");
    cie.illD50 = cie.illD50(:,2);
    cie.illD65 = load("CIE_IllD65_380-780-5nm.txt");
    cie.illD65 = cie.illD65(:,2);
    cie.illE = ones(81,1);
    cie.illF = load("CIE_IllF_1-12_380-780-5nm.txt");
    cie.illF = cie.illF(:,2:13);
    cie.eigD = load("CIE_eigD_380-780-5nm.txt");
    cie.eigD = cie.eigD(:,2:4);
    cie.checker = load("ColorChecker_380_780_5nm.txt");
```

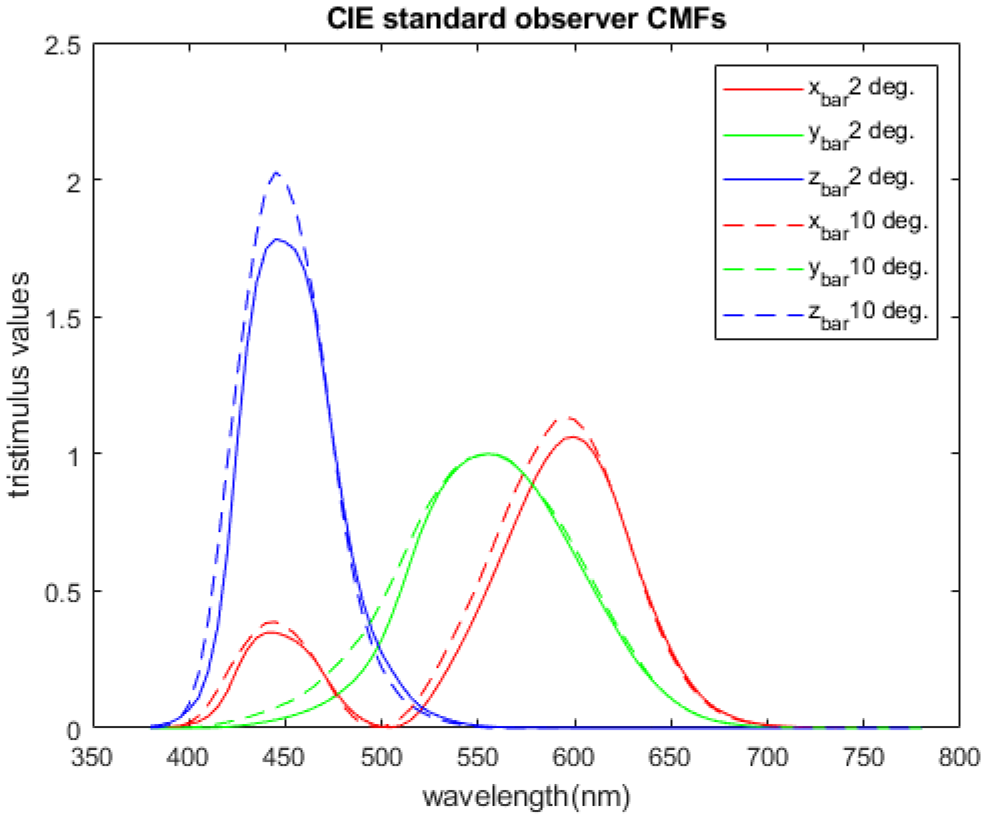
```
    cie.checker = cie.checker(:,2:25);
end
```

Step 3: Blackbody - Function

```
figure;
plot(cie.lambda, cie.cmf2deg(:,1), 'r',...
cie.lambda, cie.cmf2deg(:,2), 'g',...
cie.lambda, cie.cmf2deg(:,3), 'b',...
cie.lambda, cie.cmf10deg(:,1), '--r',...
cie.lambda, cie.cmf10deg(:,2), '--g',...
cie.lambda, cie.cmf10deg(:,3), '--b');

title('CIE standard observer CMFs')
xlabel('wavelength(nm)');
ylabel('tristimulus values')

legend('x_{bar}2 deg.',...
    'y_{bar}2 deg.',...
    'z_{bar}2 deg.',...
    'x_{bar}10 deg.',...
    'y_{bar}10 deg.',...
    'z_{bar}10 deg.');

figure;
plot(cie.lambda, cie.illA/100, '--k',...
    cie.lambda, cie.illD50/100, '--r',...
    cie.lambda, cie.illD65/100, '--b',...
    cie.lambda, blackbody(2856,380:5:780), 'k',...
    cie.lambda, blackbody(5003,380:5:780), 'r',...
    cie.lambda, blackbody(6504,380:5:780), 'b');

ylim([0 2.5]);

title('blackbody and standard illuminant spectra');
xlabel('wavelength(nm)');
ylabel('relative power');

legend('blackbody (2856K)',...
    'blackbody (5003K)',...
    'blackbody (6504K)',...
    'illuminant A',...
    'illuminant D50',...
    'illuminant D65',...
    'Location','northwest');
```

Step 3: Blackbody - Graphs





Step 4: XYZ Trisimulus Values - Function

```
% INPUT:
% 'refs' an (nx1) vector of reflectance factor data
% 'cmfs' an (nx3) set of CIE color matching functions
% 'illum' an (nx1) spectral power distribution of a light source
%
% OUTPUT:
% 'XYZ' a (3x1) vector of CIE XYZ tristimulus values
%
function XYZ = ref2XYZ(refs, cmfs, illum)

% initialize vector (3x1)
XYZ = zeros(3,1);

% calculate k value in equation: k = 100/sum(y_bar * Spectral)dLambda
% use for loop to calculate summation
sum = 0;
for n = 1:length(cmfs)
    sum = sum + (cmfs(n,2)*illum(n));
end
k = 100/sum;
```

```
% calculate X value in equation: X = k * sum(x_bar * Spectral *
 Reflectance)dLambda
% use for loop to calculate summation
sum = 0;
for n = 1:length(cmfs)
    sum = sum + (cmfs(n,1)*illum(n)*refs(n));
end
XYZ(1,1) = k*sum;

% calculate Y value in equation: Y = k * sum(y_bar * Spectral *
 Reflectance)dLambda
% use for loop to calculate summation
sum = 0;
for n = 1:length(cmfs)
    sum = sum + (cmfs(n,2)*illum(n)*refs(n));
end
XYZ(2,1) = k*sum;

% calculate Z value in equation: Z = k * sum(z_bar * Spectral *
 Reflectance)dLambda
% use for loop to calculate summation
sum = 0;
for n = 1:length(cmfs)
    sum = sum + (cmfs(n,3)*illum(n)*refs(n));
end
XYZ(3,1) = k*sum;

end
```

Step 5: ColorChecker Tristimulus Values - Results

```
>> for num = 1:24
CC_XYZs(:,num)=ref2XYZ(cie.checker(:,num),cie.cmf2deg,cie.illD65);
end

>> CC_XYZs

CC_XYZs =

             Columns 1 through 7

             11.5145    39.1346    18.3488    11.1492    25.8437    31.7110    37.1457
             10.3819    36.5981    19.6332    13.8551    24.3868    43.8600    29.5592
             7.1502    27.0564    35.6470     7.4267    45.6142    44.8778     6.5006

             Columns 8 through 14

             13.8627    29.1328     8.5889    33.9174    46.1864     8.9183    15.0353
             12.3179    19.8475     6.4569    44.1533    42.4957     6.4177    24.1079
             39.3093    14.9941    15.4745    11.4297     8.6771    32.2736     9.6379

             Columns 15 through 21
```

```
19.3447    55.8457    29.6768    14.4138    87.8402    57.9621    35.2286
11.3576    58.9726    19.3515    19.9750    92.3781    61.0426    37.0414
5.5526     9.6411    32.2626    39.0008    95.6125    65.4909     40.2256

Columns 22 through 24

19.3492     8.7646     3.2111
20.4708     9.2915     3.3763
22.1545    10.3188     3.9312
```

Step 6: X,Y Chromaticity Coordinates - Function

```
% INPUT:
% 'XYZ' a (3xn) vector of XYZ tristimulus values
%
% OUTPUT:
% 'xyY' a (3xn) vector of chromaticity coordinates (x,y) and luminance
 factor
%
function xyY = XYZ2xyY(XYZ)

% initialize output to same size as input
xyY = zeros(size(XYZ));
[r,c] = size(XYZ);

for n = 1:c
    % x = X/(X+Y+Z)
    xyY(1,n) = XYZ(1,n)/(XYZ(1,n)+XYZ(2,n)+XYZ(3,n));

    % y = Y/(X+Y+Z)
    xyY(2,n) = XYZ(2,n)/(XYZ(1,n)+XYZ(2,n)+XYZ(3,n));

    % luminance factor remains the same
    xyY(3,n) = XYZ(2,n);
end

end
```

Step 7: x,y Chromaticity Coordinates for ColorChecker chart - Results

```
>> CC_xyYs=XYZ2xyY(CC_XYZs)

CC_xyYs =

             Columns 1 through 7

             0.3964     0.3807     0.2492     0.3438     0.2696     0.2633     0.5074
             0.3574     0.3561     0.2667     0.4272     0.2544     0.3641     0.4038
```

```
10.3819   36.5981   19.6332   13.8551   24.3868   43.8600   29.5592

Columns 8 through 14

0.2117    0.4554    0.2814    0.3790    0.4744    0.1873    0.3082
0.1881    0.3102    0.2116    0.4933    0.4365    0.1348    0.4942
12.3179   19.8475    6.4569   44.1533   42.4957    6.4177   24.1079

Columns 15 through 21

0.5336    0.4487    0.3651    0.1964    0.3185    0.3142    0.3132
0.3133    0.4738    0.2381    0.2722    0.3349    0.3309    0.3293
11.3576   58.9726   19.3515   19.9750   92.3781   61.0426  37.0414

Columns 22 through 24

0.3122    0.3089    0.3053
0.3303    0.3275    0.3210
20.4708    9.2915    3.3763
```

Step 8: Interpolated Color Patch Data - Loading Function

```
% load the CIE observer and illuminant data
% define ColorMunki/Argyll/spotread measurement wavelengths
cm_lams = 380:10:730;

% define header offsets for reading the .sp files
cm_h_offset = 19;

% load and normalize the measured spectral data for the patch #1
data = importdata('28.1_real.sp', ' ', cm_h_offset);
p28_1.real = data.data/100;
data = importdata('28.1_imaged.sp', ' ', cm_h_offset);
p28_1.imaged = data.data/100;
data = importdata('28.1_matching.sp', ' ', cm_h_offset);
p28_1.matching = data.data/100;

% repeat the section above for patch #2
data = importdata('28.2_real.sp', ' ', cm_h_offset);
p28_2.real = data.data/100;
data = importdata('28.2_imaged.sp', ' ', cm_h_offset);
p28_2.imaged = data.data/100;
data = importdata('28.2_matching.sp', ' ', cm_h_offset);
p28_2.matching = data.data/100;

% interpolate for patch #1
p28_1.interp_real =
 interp1(380:10:730,p28_1.real,cie.lambda(:),'linear','extrap');
p28_1.interp_imaged =
 interp1(380:10:730,p28_1.imaged,cie.lambda(:),'linear','extrap');
p28_1.interp_matching =
 interp1(380:10:730,p28_1.matching,cie.lambda(:),'linear','extrap');
```

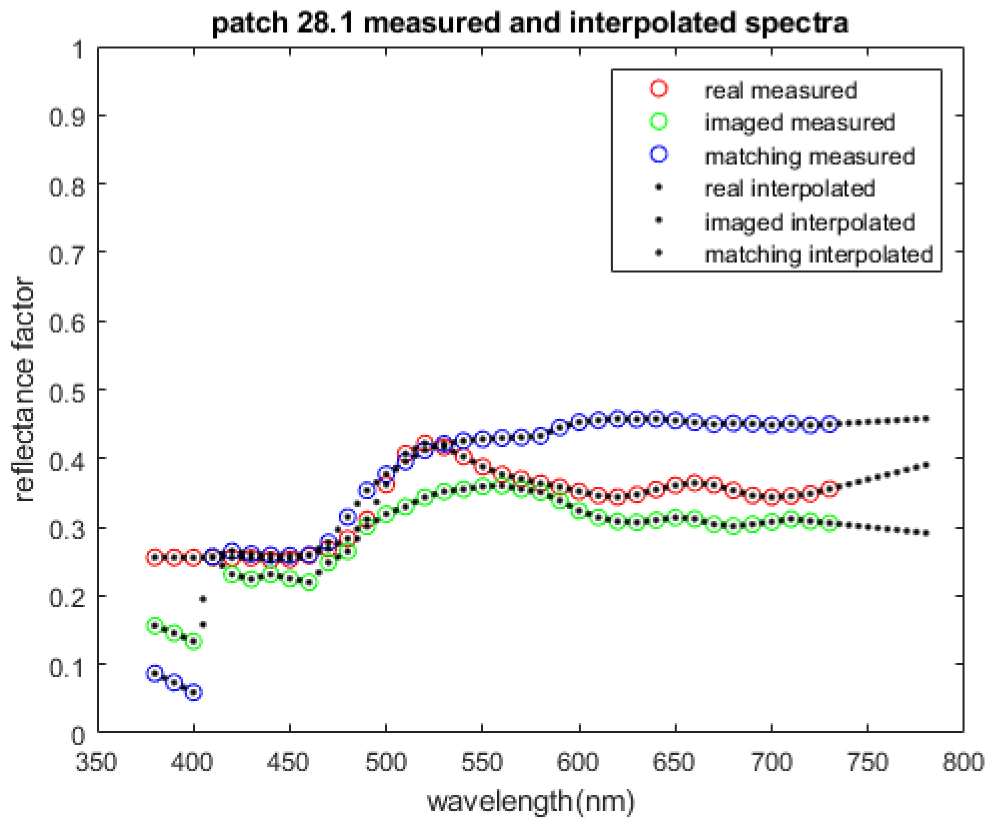
```
% repeat the section above for patch #2
p28_2.interp_real =
 interp1(380:10:730,p28_2.real,cie.lambda(:),'linear','extrap');
p28_2.interp_imaged =
 interp1(380:10:730,p28_2.imaged,cie.lambda(:),'linear','extrap');
p28_2.interp_matching =
 interp1(380:10:730,p28_2.matching,cie.lambda(:),'linear','extrap');
```

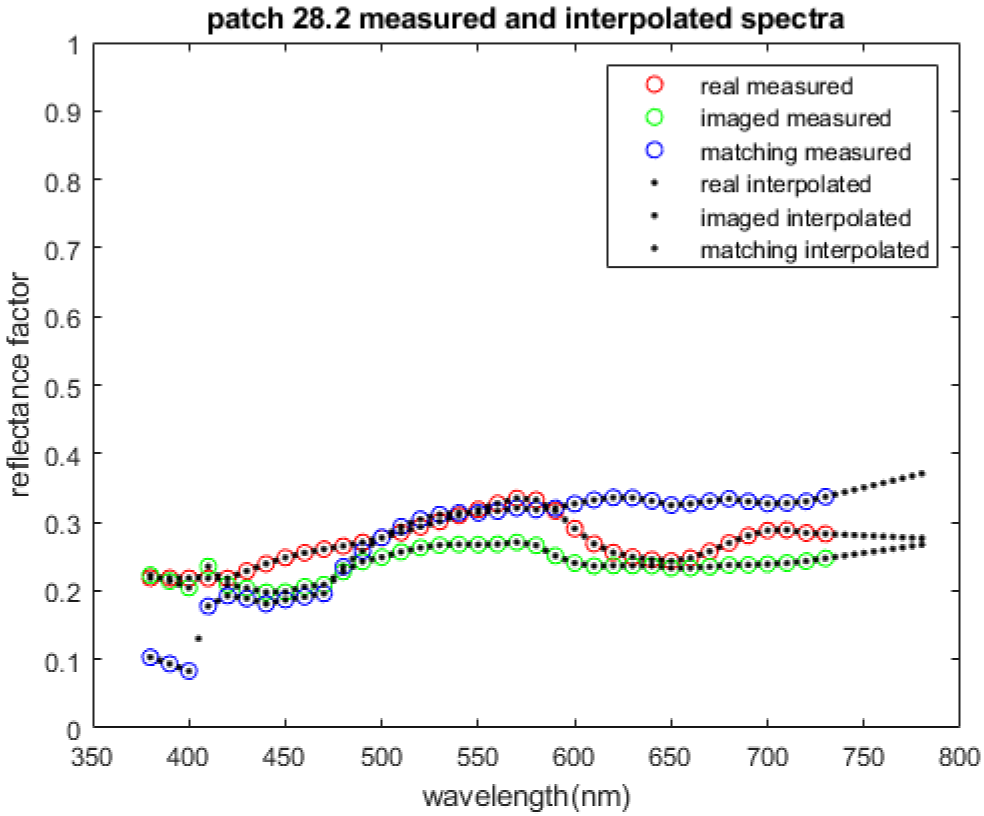
Step 9: Interpolated Color Patch Data - Plotting Function

```
figure;
plot(380:10:730, p28_1.real, 'or',...
380:10:730, p28_1.imaged, 'og',...
380:10:730, p28_1.matching, 'ob',...
cie.lambda, p28_1.interp_real, '.k',...
cie.lambda, p28_1.interp_imaged, '.k',...
cie.lambda, p28_1.interp_matching, '.k');

title('patch 28.1 measured and interpolated spectra')
xlabel('wavelength(nm)');
ylabel('reflectance factor')

legend('real measured',...
    'imaged measured',...
    'matching measured',...
    'real interpolated',...
    'imaged interpolated',...
    'matching interpolated');

ylim([0 1])

figure;
plot(380:10:730, p28_2.real, 'or',...
380:10:730, p28_2.imaged, 'og',...
380:10:730, p28_2.matching, 'ob',...
cie.lambda, p28_2.interp_real, '.k',...
cie.lambda, p28_2.interp_imaged, '.k',...
cie.lambda, p28_2.interp_matching, '.k');

title('patch 28.2 measured and interpolated spectra')
xlabel('wavelength(nm)');
ylabel('reflectance factor')

legend('real measured',...
    'imaged measured',...
    'matching measured',...
    'real interpolated',...
    'imaged interpolated',...
```

```
    'matching interpolated');
ylim([0 1])
```

Step 9: Interpolated Color Patch Data - Graphs





Step 10: Measured and Calculated Tristimulus Values for Color Patch Data - Function

```
% import data from project 2 files
L28_XYZ.real = importdata('28_XYZ_Labs_real.txt','\t', 3);
L28_XYZ.imaged = importdata('28_XYZ_Labs_imaged.txt','\t', 3);
L28_XYZ.matching = importdata('28_XYZ_Labs_matching.txt','\t', 3);

%%%%%%%%%%%%%%%%%%%%% parsing data 28.1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%

% parsing data for 28.1 real
L28_1_XYZ.real = split(L28_XYZ.real(2,:));
L28_1_XYZ.real = L28_1_XYZ.real';
L28_1_XYZ.real = str2double(L28_1_XYZ.real(:,2:4));

% parsing data for 28.1 imaged
L28_1_XYZ.imaged = split(L28_XYZ.imaged(2,:));
L28_1_XYZ.imaged = L28_1_XYZ.imaged';
L28_1_XYZ.imaged = str2double(L28_1_XYZ.imaged(:,2:4));

% parsing data for 28.1 matching
L28_1_XYZ.matching = split(L28_XYZ.matching(2,:));
L28_1_XYZ.matching = L28_1_XYZ.matching';
```

```
L28_1_XYZ.matching = str2double(L28_1_XYZ.matching(:,2:4));

%%%%%%%%%%%%%%%%%%%%% parsing data 28.2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% parsing data for 28.2 real
L28_2_XYZ.real = split(L28_XYZ.real(3,:));
L28_2_XYZ.real = L28_2_XYZ.real';
L28_2_XYZ.real = str2double(L28_2_XYZ.real(:,2:4));

% parsing data for 28.2 imaged
L28_2_XYZ.imaged = split(L28_XYZ.imaged(3,:));
L28_2_XYZ.imaged = L28_2_XYZ.imaged';
L28_2_XYZ.imaged = str2double(L28_2_XYZ.imaged(:,2:4));

% parsing data for 28.2 matching
L28_2_XYZ.matching = split(L28_XYZ.matching(3,:));
L28_2_XYZ.matching = L28_2_XYZ.matching';
L28_2_XYZ.matching = str2double(L28_2_XYZ.matching(:,2:4));


%%%%%%%%%%%%%%%%%%%%% calculating data %%%%%% %%%%%%%%%%%%%%%%%%%%%%%
%%%%%

% calculating patch 28.1 data
C28_1_XYZ.real = ref2XYZ(p28_1.interp_real, cie.cmf2deg, cie.illD50);
C28_1_XYZ.imaged = ref2XYZ(p28_1.interp_imaged, cie.cmf2deg,
 cie.illD50);
C28_1_XYZ.matching = ref2XYZ(p28_1.interp_matching, cie.cmf2deg,
 cie.illD50);

% calculating patch 28.1 data
C28_2_XYZ.real = ref2XYZ(p28_2.interp_real, cie.cmf2deg, cie.illD50);
C28_2_XYZ.imaged = ref2XYZ(p28_2.interp_imaged, cie.cmf2deg,
 cie.illD50);
C28_2_XYZ.matching = ref2XYZ(p28_2.interp_matching, cie.cmf2deg,
 cie.illD50);

fprintf("Measured and calculated tristimulus values\n\n");

fprintf("                               patch 28.1\n");
fprintf("                   measured                 calculated\n");
fprintf("             X        Y        Z        X        Y        Z
\n");
fprintf("    real "); fprintf(" %.4f ", L28_1_XYZ.real);
 fprintf(" %.4f ", C28_1_XYZ.real);      fprintf("\n");
fprintf("  imaged "); fprintf(" %.4f ", L28_1_XYZ.imaged);
 fprintf(" %.4f ", C28_1_XYZ.imaged);    fprintf("\n");
fprintf("matching ");   fprintf(" %.4f ", L28_1_XYZ.matching);
 fprintf(" %.4f ", C28_1_XYZ.matching);  fprintf("\n");

fprintf("\n");

fprintf("                               patch 28.2\n");
fprintf("                   measured                 calculated\n");
```

```
fprintf("                X         Y         Z         X         Y         Z
\n");
fprintf("    real "); fprintf(" %.4f ", L28_2_XYZ.real);
 fprintf(" %.4f ", C28_2_XYZ.real);      fprintf("\n");
fprintf("  imaged "); fprintf(" %.4f ", L28_2_XYZ.imaged);
 fprintf(" %.4f ", C28_2_XYZ.imaged);    fprintf("\n");
fprintf("matching ");   fprintf(" %.4f ", L28_2_XYZ.matching);
 fprintf(" %.4f ", C28_2_XYZ.matching);  fprintf("\n");
```

Step 10: Measured and Calculated Tristimulus Values for Color Patch Data - Results

```
Measured and calculated tristimulus values

                                      patch 28.1
                        measured                            calculated
                X         Y         Z         X         Y         Z
    real   33.5105   37.4668   22.3492   33.5174   37.4558   22.3734
  imaged   30.6663   33.8695   20.0319   30.6668   33.8586   20.0524
matching   40.5269   42.5106   23.0918   40.5229   42.5021   23.1087

                                      patch 28.2
                        measured                            calculated
                X         Y         Z         X         Y         Z
    real   27.5838   30.1686   20.7248   27.5781   30.1587   20.7281
  imaged   23.5066   25.6810   17.4180   23.5086   25.6753   17.4356
matching   29.5487   31.1759   16.7052   29.5469   31.1693   16.7208
```

Step 11: Measured and Calculated Chromaticity Coordinates for Color Patch Data - Function

```
L28_1_xyY.real = XYZ2xyY(L28_1_XYZ.real');
L28_1_xyY.imaged = XYZ2xyY(L28_1_XYZ.imaged');
L28_1_xyY.matching = XYZ2xyY(L28_1_XYZ.matching');

L28_2_xyY.real = XYZ2xyY(L28_2_XYZ.real');
L28_2_xyY.imaged = XYZ2xyY(L28_2_XYZ.imaged');
L28_2_xyY.matching = XYZ2xyY(L28_2_XYZ.matching');

C28_1_xyY.real = XYZ2xyY(C28_1_XYZ.real);
C28_1_xyY.imaged = XYZ2xyY(C28_1_XYZ.imaged);
C28_1_xyY.matching = XYZ2xyY(C28_1_XYZ.matching);

C28_2_xyY.real = XYZ2xyY(C28_2_XYZ.real);
C28_2_xyY.imaged = XYZ2xyY(C28_2_XYZ.imaged);
C28_2_xyY.matching = XYZ2xyY(C28_2_XYZ.matching);
```

```
fprintf("Measured and calculated chromaticity coordinates\n\n");

fprintf("                           patch 28.1\n");
fprintf("                 measured               calculated\n");
fprintf("            x       y       Y       x       y       Y\n");
fprintf("    real "); fprintf(" %.4f ", L28_1_xyY.real);
 fprintf(" %.4f ", C28_1_xyY.real);      fprintf("\n");
fprintf("  imaged "); fprintf(" %.4f ", L28_1_xyY.imaged);
 fprintf(" %.4f ", C28_1_xyY.imaged);    fprintf("\n");
fprintf("matching ");   fprintf(" %.4f ", L28_1_xyY.matching);
 fprintf(" %.4f ", C28_1_xyY.matching);  fprintf("\n");

fprintf("\n");

fprintf("                           patch 28.2\n");
fprintf("                 measured               calculated\n");
fprintf("            x       y       Y       x       y       Y\n");
fprintf("    real "); fprintf(" %.4f ", L28_2_xyY.real);
 fprintf(" %.4f ", C28_2_xyY.real);      fprintf("\n");
fprintf("  imaged "); fprintf(" %.4f ", L28_2_xyY.imaged);
 fprintf(" %.4f ", C28_2_xyY.imaged);    fprintf("\n");
fprintf("matching ");   fprintf(" %.4f ", L28_2_xyY.matching);
 fprintf(" %.4f ", C28_2_xyY.matching);  fprintf("\n");
```

Step 11: Measured and Calculated Chromaticity Coordinates for Color Patch Data - Results

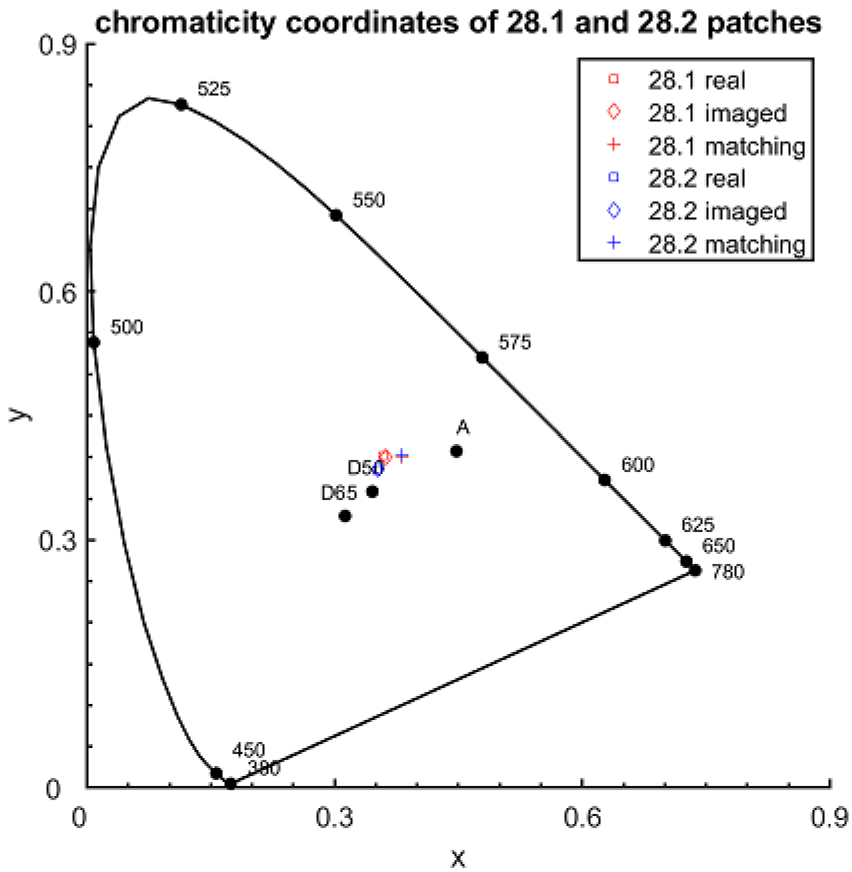
```
Measured and calculated chromaticity coordinates

                                    patch 28.1
                          measured               calculated
                     x       y       Y       x       y       Y
             real  0.3591  0.4015  37.4668  0.3591  0.4013  37.4558
           imaged  0.3626  0.4005  33.8695  0.3626  0.4003  33.8586
         matching  0.3819  0.4006  42.5106  0.3818  0.4005  42.5021

                                    patch 28.2
                          measured               calculated
                     x       y       Y       x       y       Y
             real  0.3515  0.3844  30.1686  0.3515  0.3844  30.1587
           imaged  0.3529  0.3856  25.6810  0.3529  0.3854  25.6753
         matching  0.3816  0.4026  31.1759  0.3816  0.4025  31.1693
```

Step 12: Chromaticity Diagram for Color Patch Data - Function

```
plot_chrom_diag_skel;
```

```
title('chromaticity coordinates of 28.1 and 28.2 patches');

r1 = plot(C28_1_xyY.real(1,:),C28_1_xyY.real(2,:),'sr');
i1 = plot(C28_1_xyY.imaged(1,:),C28_1_xyY.imaged(2,:),'dr');
m1 = plot(C28_1_xyY.matching(1,:),C28_1_xyY.matching(2,:),'+r');
r2 = plot(C28_2_xyY.real(1,:),C28_2_xyY.real(2,:),'sb');
i2 = plot(C28_2_xyY.imaged(1,:),C28_2_xyY.imaged(2,:),'db');
m2 = plot(C28_2_xyY.matching(1,:),C28_2_xyY.matching(2,:),'+b');


legend([r1 i1 m1 r2 i2 m2],{'28.1 real','28.1 imaged','28.1
 matching','28.2 real','28.2 imaged','28.2 matching'});
```

Step 12: Chromaticity Diagram for Color Patch Data - Graph



Step 13: Feedback

Who did what parts of the project:

- Kevin - Step 2

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

Any problems you had with the project:

- Unclear where certain variables were supposed to come from
- Not knowing that publishing needed to be a separate file from all the other files/steps

Any parts of the project you thought were valuable:

- Learning how to create and edit graphs
- Learning how to set up functions and call other functions

Published with MATLAB® R2018a