# Project 7: Color Reproduction. Team 14 (Jenee L & Justin P.)

## Table of Contents


Fetch ColorMunki XYZ and LAB data for color checker ........ 1
Evaluate color error in model ........ 2
Uncalibrated color imaging workflow ........ 3
Find errors in calibrated color imaging workflow ........ 5
Color Accurate Imaging - Colorchecker Chart Image ........ 8
display relevant functions for report ........ 8


# Fetch ColorMunki XYZ and LAB data for color checker

```
%%Multiply munki XYZs by display model

colorMunkiData = importdata('munki_CC_XYZs_Labs.txt');
munkiXYZs = colorMunkiData(:,2:4);
munkiLABs = colorMunkiData(:,5:7);

cam_RGBs = importdata('new_rgbs.mat')';

dispModel = importdata('display_model.mat');

fprintf('camera RGBs:\n');
disp(cam_RGBs);

fprintf('Reverse Display Model:\n');
disp(dispModel);
```

```
camera RGBs:
  Columns 1 through 13

    89   172    77    66   104    79   159    49   152    59   109
 154    39
    66   126    94    78    97   144    85    57    58    39   134
 106    40
    67   110   122    60   134   127    54   113    74    70    61
 54    94

  Columns 14 through 24

    49   127   170   129    35   183   135    91    61    41    32
    96    32   138    48    76   180   134    91    62    42    32
    55    48    53    88    98    69   129    91    64    47    39
```

```
Reverse Display Model:
    BLUT_disp: [1x1024 double]
    GLUT_disp: [1x1024 double]
       M_disp: [3x3 double]
    RLUT_disp: [1x1024 double]
    XYZk_disp: [0.1419 0.1358 0.2691]
```

# Evaluate color error in model

```
%Push through matrix model and conver to doubles
RGBCoords = derriveRGBs(munkiXYZs, dispModel);

%Scale RGB coordinates to a 0-100 range
RGBCoords = RGBCoords * (100/255);

dataSet = vertcat(RGBCoords, repmat(0,3,3), repmat(100,3,3));

dataReadings = [1:30; dataSet'];

%Write data to formatted ti1 file
writeTiFile('data/disp_model_test.ti1', dataReadings);

%Load patch values into workspace
disp_XYZs = importdata('disp_model_test.ti3',' ',20);

%Extract XYZ value for measured patches
munki_patch_XYZs = disp_XYZs.data(1:24,5:7);

%Generate averages of XYZ black and white values
disp_black = mean(disp_XYZs.data(25:27,5:7));
disp_white = mean(disp_XYZs.data(28:30,5:7));

display_labs = XYZ2Lab(munki_patch_XYZs',disp_white)';

%Generate delta EAB values for display vs colormunki readings
display_deltas = deltaEab(display_labs', munkiLABs');

%Construct display table of LAB values
step1_table = [munkiLABs, display_labs, display_deltas'];
step1_table = ([1:24; step1_table'])';

fprintf('step1_table:\n');
disp(step1_table);

csvwrite('data/step1_out.csv', step1_table);
```

```
step1_table:
  Columns 1 through 7

    1.0000   37.1865   14.9985   15.2592   36.5125   13.6499   22.0570
    2.0000   65.8188   16.8695   18.0267   65.0199   13.0576   34.4751
    3.0000   49.9949   -3.1841  -23.5159   50.1295   -6.6342    0.5711
```

```
    4.0000   42.6411  -15.3251   20.0423   43.3237  -17.0846   31.0136
    5.0000   54.6852    9.6978  -26.7126   53.6134    7.4958   -4.1603
    6.0000   71.2441  -33.1391   -0.5010   70.6834  -32.4789   21.2565
    7.0000   62.2558   34.1094   57.7774   61.0289   33.6489   58.8666
    8.0000   39.5890    9.9980  -43.6388   38.7346    6.8452  -20.6578
    9.0000   51.8424   48.1403   16.0636   49.5325   46.4403   26.6212
   10.0000   29.4495   22.4255  -21.7661   28.5576   19.2583   -5.2784
   11.0000   71.6264  -24.3441   57.6850   71.0990  -23.8198   64.0447
   12.0000   72.2288   20.6039   69.0149   71.4147   20.6359   70.2883
   13.0000   28.6402   18.5907  -51.4092   28.6664   12.2505  -27.4617
   14.0000   54.6309  -39.5493   32.8341   55.4049  -38.1206   44.1093
   15.0000   42.5988   54.6049   25.7315   40.8986   52.5362   29.0832
   16.0000   82.4265    3.8689   78.8570   81.5904    4.2572   79.7039
   17.0000   51.5476   49.5154  -14.3758   49.3202   47.2057    3.4835
   18.0000   49.3892  -26.5473  -28.6645   49.3182  -27.9416   -3.9655
   19.0000   95.4458   -0.4414    0.0244   94.4964   -3.7891   25.6864
   20.0000   80.0339    0.1309   -0.9345   78.8750   -1.6359   22.4232
   21.0000   66.0107   -0.0004   -1.1463   64.8494   -0.8979   19.5111
   22.0000   50.5546   -0.6207   -0.9616   50.1380   -2.8325   17.9094
   23.0000   35.1532   -0.0632   -0.9708   34.7158   -1.5069   13.2715
   24.0000   20.3224   -0.2858   -0.5603   22.0298   -1.4769    7.6320

  Column 8

    6.9630
   16.9032
   24.3332
   11.1325
   22.6849
   21.7747
    1.7040
   23.2120
   10.9402
   16.8129
    6.4030
    1.5117
   24.7725
   11.3916
    4.2899
    1.2518
   18.1452
   24.7384
   25.8968
   23.4530
   20.7095
   19.0048
   14.3220
    8.4527
```

# Uncalibrated color imaging workflow

```
%Fetch average RGBs of chart from project 5
```

```
% Using Readings from Jenee's 15" 2010 matte Macbook Pro

%Scale RGB values
cam_RGBs_scale = double(cam_RGBs);
cam_RGBs_scale = cam_RGBs_scale .* (100/255);

%Construct matrix to write to ti1 file for colormunki
dataSet = [cam_RGBs_scale, repmat(0,3,3),repmat(100,3,3)];
dataSet = [1:30;dataSet];

%Write data to formatted ti1 file
writeTiFile('data/workflow_test_uncal.ti1', dataReadings);

uncal_XYZs = importdata('workflow_test_uncal.ti3',' ',20);

%Extract XYZ value for measured patches
munki_patch_XYZs = uncal_XYZs.data(1:24,5:7);

%Generate averages of XYZ black and white values
disp_black = mean(uncal_XYZs.data(25:27,5:7));
disp_white = mean(uncal_XYZs.data(28:30,5:7));

display_labs = XYZ2Lab(munki_patch_XYZs',disp_white)';

%Generate delta EAB values for display vs colormunki readings
display_deltas = deltaEab(display_labs', munkiLABs');

%Construct display table of LAB values
step2_table = [munkiLABs, display_labs, display_deltas'];
step2_table = ([1:24; step2_table'])';

fprintf('step2_table:\n');
disp(step2_table);

csvwrite('data/step2_out.csv', step2_table);
```

*step2_table:*

*Columns 1 through 7*

| | | | | | | |
|---|---|---|---|---|---|---|
| 1.0000 | 37.1865 | 14.9985 | 15.2592 | 36.3901 | 13.5036 | 21.8902 |
| 2.0000 | 65.8188 | 16.8695 | 18.0267 | 64.8938 | 12.9147 | 34.3872 |
| 3.0000 | 49.9949 | -3.1841 | -23.5159 | 50.0694 | -6.6281 | 0.5147 |
| 4.0000 | 42.6411 | -15.3251 | 20.0423 | 43.2657 | -17.1453 | 30.9406 |
| 5.0000 | 54.6852 | 9.6978 | -26.7126 | 53.5525 | 7.4835 | -4.2158 |
| 6.0000 | 71.2441 | -33.1391 | -0.5010 | 70.6336 | -32.4962 | 21.2360 |
| 7.0000 | 62.2558 | 34.1094 | 57.7774 | 60.9524 | 33.5602 | 58.7934 |
| 8.0000 | 39.5890 | 9.9980 | -43.6388 | 38.6788 | 6.9239 | -20.6984 |
| 9.0000 | 51.8424 | 48.1403 | 16.0636 | 49.4612 | 46.3333 | 26.5766 |
| 10.0000 | 29.4495 | 22.4255 | -21.7661 | 28.5173 | 19.1934 | -5.3256 |
| 11.0000 | 71.6264 | -24.3441 | 57.6850 | 71.0378 | -23.9541 | 64.0276 |
| 12.0000 | 72.2288 | 20.6039 | 69.0149 | 71.3531 | 20.5486 | 70.2778 |
| 13.0000 | 28.6402 | 18.5907 | -51.4092 | 28.6255 | 12.3484 | -27.4836 |
| 14.0000 | 54.6309 | -39.5493 | 32.8341 | 55.3623 | -38.1732 | 44.0659 |
| 15.0000 | 42.5988 | 54.6049 | 25.7315 | 40.8206 | 52.4161 | 28.9956 |

```
   16.0000   82.4265    3.8689   78.8570   81.5382    4.1630   79.7412
   17.0000   51.5476   49.5154  -14.3758   49.2470   47.1476    3.3989
   18.0000   49.3892  -26.5473  -28.6645   49.2791  -27.8250   -3.9755
   19.0000   95.4458   -0.4414    0.0244   94.4708   -3.8201   25.6692
   20.0000   80.0339    0.1309   -0.9345   78.8230   -1.7004   22.3735
   21.0000   66.0107   -0.0004   -1.1463   64.7895   -0.9669   19.4667
   22.0000   50.5546   -0.6207   -0.9616   50.0684   -2.9007   17.8546
   23.0000   35.1532   -0.0632   -0.9708   34.6589   -1.5631   13.2275
   24.0000   20.3224   -0.2858   -0.5603   21.9760   -1.5251    7.6228

  Column 8

    6.8439
   16.8571
   24.2763
   11.0669
   22.6339
   21.7551
    1.7415
   23.1633
   10.9297
   16.7812
    6.3818
    1.5378
   24.7266
   11.3394
    4.3136
    1.2874
   18.0786
   24.7223
   25.8848
   23.4111
   20.6717
   18.9601
   14.2859
    8.4400
```

# Find errors in calibrated color imaging workflow

```
%Estimate XYZs of patches via camera model

%Import lookup tables and forward matrix
RLUT_fwd = importdata('RLUT_fwd.mat');
GLUT_fwd = importdata('GLUT_fwd.mat');
BLUT_fwd = importdata('BLUT_fwd.mat');
M_fwd = importdata('M_fwd.mat');

cam_LUT_Results = zeros(3, 24);

% push the DCs through the forward LUTs to predict radiometric scalars
```

```
for i=1:size(cam_RGBs, 2)
    cam_LUT_Results(1,i) = RLUT_fwd(cam_RGBs(1,i)+1);
    cam_LUT_Results(2,i) = GLUT_fwd(cam_RGBs(2,i)+1);
    cam_LUT_Results(3,i) = BLUT_fwd(cam_RGBs(3,i)+1);
end

% add the homogeneous coordinate required to apply the forward matrix
cam_RSs_h = [cam_LUT_Results; ones(1,size(cam_LUT_Results,2))];

% apply the forward matrix to the RSs to calculate model-predicted
 XYZs
modeled_XYZs = M_fwd * cam_RSs_h * 100;

%convert XYZs from D50 to D65
cie = loadCIEData();
D50_XYZ = ref2XYZ(cie.illE, cie.cmf2deg, cie.illD50);
D65_XYZ = ref2XYZ(cie.illE, cie.cmf2deg, cie.illD65);
modeled_XYZs = catBradford(modeled_XYZs, D50_XYZ, D65_XYZ);

%Pass new XYZs through display model to convert back to RGBs
modeled_RGBs = derriveRGBs(modeled_XYZs', dispModel);

%Rescale values into a 0-100 range
modeled_RGBs = modeled_RGBs .* (100/255);

%Construct matrix to write to ti1 file for colormunki
modeled_RGBs = [modeled_RGBs', repmat(0,3,3),repmat(100,3,3)];
modeled_RGBs = [1:30;modeled_RGBs];

%Write data to file to be parsed by colormunki
writeTiFile('data/workflow_test_cal.ti1', modeled_RGBs);

cal_XYZs = importdata('workflow_test_cal.ti3',' ',20);

%Extract XYZ value for measured patches
munki_patch_XYZs = cal_XYZs.data(1:24,5:7);

%Generate averages of XYZ black and white values
disp_black = mean(cal_XYZs.data(25:27,5:7));
disp_white = mean(cal_XYZs.data(28:30,5:7));

display_labs = XYZ2Lab(munki_patch_XYZs',disp_white)';

%Generate delta EAB values for display vs colormunki readings
display_deltas = deltaEab(display_labs', munkiLABs');

%Construct display table of LAB values
step3_table = [munkiLABs, display_labs, display_deltas'];
step3_table = ([1:24; step3_table'])';

fprintf('step3_table:\n');
disp(step3_table);

csvwrite('data/step3_out.csv', step3_table);
```

*step3_table:*
*Columns 1 through 7*

| | | | | | | |
|---|---|---|---|---|---|---|
| 1.0000 | 37.1865 | 14.9985 | 15.2592 | 30.5034 | 9.4084 | -4.2448 |
| 2.0000 | 65.8188 | 16.8695 | 18.0267 | 57.2868 | 11.4696 | 4.9414 |
| 3.0000 | 49.9949 | -3.1841 | -23.5159 | 40.3285 | 7.0173 | -29.6032 |
| 4.0000 | 42.6411 | -15.3251 | 20.0423 | 32.6710 | -10.1759 | 4.3664 |
| 5.0000 | 54.6852 | 9.6978 | -26.7126 | 43.7451 | 17.0640 | -33.6697 |
| 6.0000 | 71.2441 | -33.1391 | -0.5010 | 56.9354 | -22.2707 | -7.8499 |
| 7.0000 | 62.2558 | 34.1094 | 57.7774 | 44.1174 | 20.4648 | 27.2331 |
| 8.0000 | 39.5890 | 9.9980 | -43.6388 | 26.4006 | 25.8724 | -45.6183 |
| 9.0000 | 51.8424 | 48.1403 | 16.0636 | 37.2194 | 36.6735 | 1.5544 |
| 10.0000 | 29.4495 | 22.4255 | -21.7661 | 19.9181 | 18.1023 | -22.4878 |
| 11.0000 | 71.6264 | -24.3441 | 57.6850 | 53.4545 | -27.7132 | 34.1255 |
| 12.0000 | 72.2288 | 20.6039 | 69.0149 | 48.6947 | 5.4684 | 33.2883 |
| 13.0000 | 28.6402 | 18.5907 | -51.4092 | 20.0301 | 27.1531 | -43.4847 |
| 14.0000 | 54.6309 | -39.5493 | 32.8341 | 37.7569 | -28.2910 | 15.3137 |
| 15.0000 | 42.5988 | 54.6049 | 25.7315 | 27.0814 | 38.5761 | 6.5510 |
| 16.0000 | 82.4265 | 3.8689 | 78.8570 | 59.5985 | -6.5902 | 47.4951 |
| 17.0000 | 51.5476 | 49.5154 | -14.3758 | 32.0570 | 40.3556 | -18.2220 |
| 18.0000 | 49.3892 | -26.5473 | -28.6645 | 31.5600 | -1.3383 | -27.1892 |
| 19.0000 | 95.4458 | -0.4414 | 0.0244 | 72.8135 | -23.9592 | 52.4417 |
| 20.0000 | 80.0339 | 0.1309 | -0.9345 | 56.7508 | -0.8568 | -8.0799 |
| 21.0000 | 66.0107 | -0.0004 | -1.1463 | 39.3275 | 0.8852 | -10.3791 |
| 22.0000 | 50.5546 | -0.6207 | -0.9616 | 27.1931 | -0.3235 | -7.4736 |
| 23.0000 | 35.1532 | -0.0632 | -0.9708 | 17.7867 | 1.3967 | -8.3134 |
| 24.0000 | 20.3224 | -0.2858 | -0.5603 | 12.7798 | 1.4626 | -7.9057 |

*Column 8*

| |
|---|
| 21.3616 |
| 16.5281 |
| 15.3155 |
| 19.2782 |
| 14.9113 |
| 19.4130 |
| 38.0544 |
| 20.7328 |
| 23.5762 |
| 10.4910 |
| 29.9436 |
| 45.3799 |
| 14.4999 |
| 26.8039 |
| 29.4212 |
| 40.1756 |
| 21.8764 |
| 30.9120 |
| 61.7485 |
| 24.3749 |
| 28.2493 |
| 24.2540 |
| 18.9114 |
| 10.6725 |

# Color Accurate Imaging - Colorchecker Chart Image

```
chartImg = importdata('chart.jpg');
chartSize = size(chartImg);
img_LUT_Results = zeros(chartSize(1), chartSize(2), chartSize(3));

% push the DCs through the forward LUTs to predict radiometric scalars
for i=1:800
    for j=1:1125
        img_LUT_Results(i,j,1) = RLUT_fwd(chartImg(i,j,1)+1);
        img_LUT_Results(i,j,2) = GLUT_fwd(chartImg(i,j,2)+1);
        img_LUT_Results(i,j,3) = BLUT_fwd(chartImg(i,j,3)+1);
    end
end

%squash image matrix into a single column for processing
img_LUT_Results = reshape(img_LUT_Results, [800 * 1125,3]);

% add the homogeneous coordinate required to apply the forward matrix
img_RSs_h = [img_LUT_Results'; ones(1,size(img_LUT_Results',2))];

% apply the forward matrix to the RSs to calculate model-predicted XYZs
img_modeled_XYZs = M_fwd * img_RSs_h * 100;

%Convert XYZ values through model back to RGB
img_RGB_Results = derriveRGBs(img_modeled_XYZs', dispModel);

%Bring image shape back to three dimensions
img_RGB_Results = reshape(img_RGB_Results, [800, 1125, 3]);

imwrite(img_RGB_Results, 'img/result.jpg');
```

# display relevant functions for report

```
dbtype('writeTiFile.m');
dbtype('derriveRGBs.m');
```

```
function [] = writeTiFile( name, dataMatrix )
    fileId = fopen(name, 'w');
    fprintf(fileId,'CTI1\n\nCOLOR_REP "RGB"\nNUMBER_OF_FIELDS 4\nBEGIN_DATA_FORMAT\n');
    fprintf(fileId,'SAMPLE_ID RGB_R RGB_G RGB_B\nEND_DATA_FORMAT\n');
    fprintf(fileId,'\nNUMBER_OF_SETS 30\nBEGIN_DATA\n');
    fprintf(fileId, '%d %3.3f %3.3f %3.3f\n', dataMatrix);
    fprintf(fileId,'END_DATA');
    fclose(fileId);
```

```
end

% Given a display model matrix and lookup tables, convert a set of
% XYZ coords using black reference XYZ into scalar RGBs

%Prerequisites - valid load_ramps_data_1516 script on path
function [ result ] = derriveRGBs( XYZs, dispModel )
    run load_ramps_data_1516;
    cie = loadCIEData();
    D50_XYZ = ref2XYZ(cie.illE, cie.cmf2deg, cie.illD50);
    D65_XYZ = ref2XYZ(cie.illE, cie.cmf2deg, cie.illD50);

    adapt_XYZs = catBradford(XYZs',D50_XYZ, D65_XYZ);

    % Subtract XYZ black from each adapted value
    adapt_sz = size(adapt_XYZs);
    adapt_XYZs = adapt_XYZs' - repmat(black_XYZ,adapt_sz(2),1);

    %Multiply by matrix to obtain radiometric scalars
    scalars = adapt_XYZs * dispModel.M_disp';

    % Normalize scalars by 100
    scalars = scalars/100;

    % Clip any out of range values
    scalars(scalars<0) = 0;
    scalars(scalars>1) = 1;

    %Multiply scalars by 1023 and round to nearest integer
    scalars = round(scalars * 1023) + 1;

    % Convert to 8 bit unsigned integers
    R_LUT_RESULT = uint8(dispModel.RLUT_disp(scalars(:,1)))';
    G_LUT_RESULT = uint8(dispModel.GLUT_disp(scalars(:,2)))';
    B_LUT_RESULT = uint8(dispModel.BLUT_disp(scalars(:,3)))';

    result = [R_LUT_RESULT G_LUT_RESULT B_LUT_RESULT];
end
```

*Published with MATLAB® R2015b*