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

Table of Contents

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);
```

Evaluate color error in model

```
dispModel = importdata('display_model.mat');

%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('disp_model_test.ti1', dataReadings);
```

Uncalibrated color imaging workflow

```
%Fetch average RGBs of chart from project 5
% Using Readings from Jenee's 15" 2010 matte Macbook Pro

cam_RGBs = importdata('cam_rgbs.mat');

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

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

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

display relevant functions for report

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

```
1     function [] = writeTiFile( name, dataMatrix )
2         fileId = fopen(name, 'w');
3         fprintf(fileId,'CTI1\n\nCOLOR_REP "RGB"\nNUMBER_OF_FIELDS
 4\nBEGIN_DATA_FORMAT\n');
4         fprintf(fileId,'SAMPLE_ID RGB_R RGB_G RGB_B\nEND_DATA_FORMAT
\n');
5         fprintf(fileId,'\nNUMBER_OF_SETS 30\nBEGIN_DATA\n');
6         fprintf(fileId, '%d %3.3f %3.3f %3.3f\n', dataMatrix);
7         fprintf(fileId,'END_DATA');
8         fclose(fileId);
9     end
10
```

```
1     % Given a display model matrix and lookup tables, convert a set
 of
2     % XYZ coords using black reference XYZ into scalar RGBs
3
4     %Prerequisites - valid load_ramps_data_1516 script on path
5     function [ result ] = derriveRGBs( XYZs, dispModel )
6         run load_ramps_data_1516;
7         cie = loadCIEData();
8         D50_XYZ = ref2XYZ(cie.illE, cie.cmf2deg, cie.illD50);
9         D65_XYZ = ref2XYZ(cie.illE, cie.cmf2deg, cie.illD50);
10
11        adapt_XYZs = catBradford(XYZs',D50_XYZ, D65_XYZ);
12
13        % Subtract XYZ black from each adapted value
14        adapt_XYZs = adapt_XYZs' - repmat(black_XYZ,24,1);
15
16        %Multiply by matrix to obtain radiometric scalars
17        scalars = adapt_XYZs * dispModel.M_disp;
18
19        % Normalize scalars by 100
20        scalars = scalars/100;
21
22        % Clip any out of range values
23        scalars(scalars<0) = 0;
24        scalars(scalars>1) = 1;
25
```

```
        %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