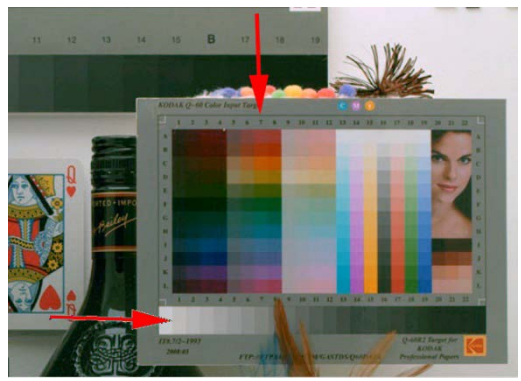
OPTI 600C Homework 2 Due: October 6, 2017

This homework involves white balancing and color calibration of an image. There are a series of steps that need to be followed closely in order to get usable results.

1. Display the image White Balance.jpg. Note that the whites do not look white in the image, but instead have a greenish tinge to them.
2. One thing we can do is white balance the image. This means basically find something known to be gray in the image and adjust the RGB channels separately so that a gray patch has equal RGB values. Convert each of the image channels to a separate array and divide by 255 to put the RGB values into the range of zero to one. Pick a pixel in one of the gray patches in the calibration target at the top of the image. Scale each channel of the image so that the RGB values for this pixel are the same. Find the peak value of the three channels and scale the arrays to a peak value of 255. Display this white balanced image.
3. The white balanced image should have grays that appear gray now, but the colors will look flat. The next step is color calibration. The image contains a Kodak Q-60 Color Input Target. This is a standardized target containing a variety of know color patched. The documentation for this chart is in the KodakQ60.pdf file. We are going to do the color calibration based on 12 of the colors and 12 of the grays in this target. The file CalibrationData.txt contains colorimetric values for column 7 lines A through L of the colors and Steps 1 through 12 of the gray scale ramp. The data are in polar CIELAB format with L^*, C^*_{ab}, h_{ab} values. Calculate the CIELAB a^* and b^* for each one and plot a^* vs. b^*. You should get a roughly circular series of uniformly spaced points. Also, calculate the corresponding X, Y and Z Tristimulus values assuming a D65 white point (X_n = 95.047, Y_n = 100, Z_n = 108.883).



4. Convert the XYZ values to linear sRGB values using the inverse of the conversion matrix from the class notes. Scale each of the linear RGB values by 0.01. This last step converts the brightness from a range of 0…100 (since Y_n = 100) to a range of 0…1. Finally, set any negative RGB values to zero. These correspond to colors that cannot be displayed in the sRGB system. The resultant RGB values will be your target values.
5. We now want to find a 3x3 conversion matrix that takes the actual RGB values from the calibration target and converts them to the target values found in the preceding step. This is easier said than done and you will need to settle for a least squares fit. We are going to solve the following matrix equation: $\mathbf{MA} = \mathbf{B}$, where $\mathbf{M}$ is the 3x3 conversion matrix, $\mathbf{A}$ is a 3x24 matrix whose columns contain the actual RGB values (scale to 0…1) of each of the patches in the image you generated in part 2, and $\mathbf{B}$ is a 3x24 matrix whose columns contain the target RGB

values you found in part 4. Create the $\mathbf{A}$ and $\mathbf{B}$ matrices and calculate the conversion matrix with $\mathbf{M} = \mathbf{B}\mathbf{A}^T\left[\mathbf{A}\mathbf{A}^T\right]^{-1}$, where *T* is the transpose operator.

6. Almost there. Scale each of the rows of $\mathbf{M}$ independently so that they sum to one. The least squares operation messes up the white balancing you did previously. By scaling each row independently so that they sum to one, a gray input value will get mapped to a gray output value.
7. Finally, for each pixel in the image, convert the RGB value to the range of 0…1 and make it a 3x1 column vector. Multiply this column vector by $\mathbf{M}$ to get your color corrected RGB value. Scale back to the range 0…255 and display your image.
8. I found one last step was needed. The reflection of the top of the watch seems to consume a decent portion of the dynamic range of the image. I adjusted the brightness so that the range from 0…200 was stretched to 0…255 and anything above 255 was clipped to 255.