Lecture Review and Quizzes (Due: Wednesday, January 24, 1:30pm)
Please review what you have learned in class and then complete the online quiz questions for the following sections on OpenEdX:\(^1\):

- Color Science
- Image Segmentation

Homework #2
Released: Monday, January 15
Due: Wednesday, January 24, 1:30pm

1. Color Reproduction (Total of 10 points)

In this problem, we study the accurate reproduction of color, exploiting the trichromacy of human vision. From the handouts webpage, please download the following MATLAB files:

- hw2_XYZ.mat: contains the CIE 1931 XYZ spectral matching functions
- hw2_face.mat: contains a hyperspectral image of a human face, where the spectral reflectance \(\rho\) is measured at each pixel (reflectance \(\rho\) is a value between 0 and 1)
- hw2_light.mat: contains the spectrum for an illuminant, scaled to result in a peak luminance of 100 cd/m\(^2\) (for reflectance \(\rho = 1\) across the spectrum) and represented in units of Watts/sr/nm/m\(^2\)
- hw2_display.mat: contains the spectra for 3 primaries in an LCD display, represented in units of Watts/sr/nm/m\(^2\)

For all files, the spectrum is sampled in 5nm increments from 370nm to 730nm.

Consider a discretized version of the three-receptor model of color perception:

\[
\begin{align*}
X &= \sum_{n=1}^{N} \bar{x}[n] \ I[n] \ \rho[n] \\
Y &= \sum_{n=1}^{N} \bar{y}[n] \ I[n] \ \rho[n] \\
Z &= \sum_{n=1}^{N} \bar{z}[n] \ I[n] \ \rho[n]
\end{align*}
\]

\(\bar{x}[n], \ \bar{y}[n], \ \bar{z}[n]\) are samples of the XYZ spectral matching functions, \(I[n]\) are samples of the illuminant’s spectrum, and \(\rho[n]\) are samples of the spectral reflectance of the illuminated surface (e.g., the human face in the file hw2_face.mat).

(a) Find the linear mapping from the spectral reflectance of the face to the weights one would apply to the three display primaries such that a human observer sees the same colors on the display as viewing the face directly under the illuminant.

\((4 \text{ points})\)

\(^1\) https://suclass.stanford.edu/courses/course-v1:Engineering+EE368+Winter2018
(b) Generate, show, and submit an RGB image, assuming the display primaries are the same as those provided in this problem and the display has a $\gamma$ of 2.2. 

(c) Generate and submit a plot of the product of the face’s spectral reflectance response and the illuminant’s spectral response at pixel position [x,y] = [47,149] in the image. (Note: A MATLAB array for a color image is indexed like Image[row, col, channel].) Then, generate and submit a plot the corresponding spectral response emitted by the display at the same pixel position. Compare the two spectra. Are the spectra metamers or isomers?

Note: Please attach relevant MATLAB code.
2. Color Gamut and Saturation (Total of 8 points)

In this problem, we study the effects of saturation on the display of different colors. From the handouts webpage, please download the following MATLAB files:

- `hw2_chroma_solid.fig`: contains a figure showing the boundaries of several cross-sections of the color gamut in XYZ space
- `hw2_chroma_map.mat`: contains a binary image marking the interior of the CIE chromaticity diagram in xy space

(a) Suppose colors from the color gamut are represented in the CIE RGB color space. In practical applications, the weights for the R, G, and B primaries must be (i) nonnegative and (ii) less than a maximum value. Suppose the individual RGB values must lie in the range [0,1] and values outside this range are clipped to 0 or 1. Perform a linear transformation of the unit cube \( \{0 \leq R \leq 1, 0 \leq G \leq 1, 0 \leq B \leq 1\} \) from the CIE RGB color space into the CIE XYZ color space. Display the edges of the resulting parallelepiped in XYZ space on the same axes as the cross-sections of the color gamut and submit the figure. In relation to the parallelepiped, which colors in the color gamut would be clipped/saturated in the RGB color space?

(4 points)

(b) Render and submit xy-chromaticity diagrams corresponding to the following cross-sections of the color gamut: \( X + Y + Z = S \) for \( S = 0.5, 1.0, 2.0 \). Assume a display \( \gamma \) of 2.2. For each chromaticity diagram, display and submit a binary image where regions of R, G, or B clipping/saturation are marked by white pixels and all other regions are marked by black pixels. Interpret the shape of the clipped/saturated regions in relation to the parallelepiped generated in (a).

(4 points)

Note: Please attach relevant MATLAB code.
3. Binarization of Scanned Book Pages (Total of 8 points)

Services like the Guttenberg Project and Internet Archive have digitized large collections of books. First, the book pages are scanned. Then, the scanned images are binarized and processed through an optical character recognition (OCR) engine. For modern books, the book spine can be removed to separate the book pages for more efficient scanning. For vintage books, however, destroying the original binding of the book is often undesirable. If a book is scanned with its spine left intact, the curvature of the pages causes uneven illumination in the resulting images. This effect can be observed in the images hw2_book_page_1.jpg and hw2_book_page_2.jpg, which are available on the handouts webpage.

(a) For each image, generate a binary image by performing global thresholding. Use a threshold chosen by Otsu’s method (function: graythresh). Submit the binary image. Also, submit a histogram of the original image’s gray values and clearly mark the threshold of Otsu’s method on this histogram. Comment on the quality of the binary image.

(4 points)

(b) Now, perform locally adaptive thresholding. Treat uniform and non-uniform regions differently based on the local variance. Submit the binary image. Comment on the quality of the binary image compared to the result from part (a).

(4 points)

Note: Please attach relevant MATLAB code.
4. Traffic Cone Detection (Total of 9 points)

Traffic cones are set up by law enforcement and construction workers to block a portion of the road. Unlike more permanent road signs whose locations are indexed in digital maps and hence available to navigation systems on automobiles, traffic cones are missing from the digital maps and must be dynamically detected. Please download the following images from the handouts webpage:

- hw2_cone_training_{1,2,3,4,5}.jpg: RGB images for training
- hw2_cone_training_map_{1,2,3,4,5}.png: binary masks of cones in the training images
- hw2_cone_testing_{1,2}.jpg: RGB images for testing

(a) Using the 5 training images, generate and submit a 3-d scatterplot in RGB space of a small randomly chosen subset of the training RGB samples, where the cone and non-cone RGB samples are plotted with different markers (function: scatter3). Comment on how effectively cone and non-cone samples are separated in RGB space.

(3 points)

(b) Using the 5 training images, train a multidimensional MAP detector in RGB space for distinguishing between traffic cones and other parts of the image. Assuming RGB values lie in the range [0,255], use 16x16x16 uniformly spaced bins for the RGB values. Report the fraction of bins in the 16x16x16 grid which are labeled as belonging to the cone class.

(3 points)

(c) Use the multidimensional MAP detector to classify pixels in each testing image as “cone” or “non-cone”. After MAP classification, you can also perform any post-processing (e.g., small region removal) to make the final result more accurate. For each testing image, submit a binary image where the “cone” pixels are shown as white and the “non-cone” pixels are shown as black.

(3 points)

Note: Please attach relevant MATLAB code.
Bonus Problems

You can get 100% of the credit on this homework without doing any of the bonus problems. You can get 4 extra points if you solve one of the two bonus problems. If you do so, please submit your solution for only one of them (otherwise only the first will be graded and we will ignore the second).

**Bonus #1: Mobile Image Processing with Android (Total of 4 points)**

This problem may be appealing to students who are planning to use Android devices in their final projects, because the problem directs you to set up the Android development environment.

Read Tutorial #1 (Basic Android Setup) on the class Android webpage. Download and install all the necessary software for basic Android programming as explained in Tutorial #1.

Run the “Viewfinder EE368 (Color Histograms)” application on an Android device or an emulator. Take a screenshot of the application running on the device or emulator, clearly showing the red, green, and blue histograms overlaid on the screen. Submit your screenshot.

(4 points)
Bonus #2: Color Balancing (Total of 4 points)
In this problem, we explore the effectiveness of some popular color balancing algorithms. From the handouts webpage, please download hw2_macbeth.jpg and hw2_leaf.jpg, which contain values in the $\gamma$-predistorted domain.

(a) Convert the images to the linear RGB domain assuming a display $\gamma$ of 2.2. Then, perform color balancing on each image in the linear RGB domain using each of the following algorithms:

1. Gray-world: Choose the scale factors $[k_r, k_g, k_b]$ so that $k_g = 1$.
2. Scale-by-max: Choose the scale factors $[k_r, k_g, k_b]$ so that after scaling, the max value in each color component in the image is 1.
3. Shades-of-gray. Choose the scale factors $[k_r, k_g, k_b]$ so that $k_g = 1$. Use $p = 6$.

Report the $[k_r, k_g, k_b]$ values and display the color-balanced images for each algorithm. Please remember to apply $\gamma$ predistortion before displaying. In which cases do the algorithms perform well? In which cases do the algorithms perform poorly? You do **not** need to submit the color-balanced images.

(2 points)

(b) Perform color balancing on each image using the shades-of-gray algorithm for the different values $p = 1, 2, 6, 10, 100$. Note that the $p = 1$ generates a result equivalent to that of gray-world, while $p = 100$ generates a result very similar to that of scale-by-max. In each case, plot and submit the histograms for each color channel after color balancing. Also plot and submit the histogram before color balancing. How are the histograms transformed through the color balancing operation for the different $p$ values? You do **not** need to submit the color-balanced images.

(2 points)

Note: Please attach relevant MATLAB code.