Name: ______________________________

EE 193-03 Homework 2

Due via `provide` at 11:59pm, 17 October 2018

Submission

Zip up your code (i.e., the "code" subdirectory of the starter code package) and submit it via provide from one of the Linux machines:

```
provide ee193HIP hw3 <ZIP FILE>
```

You can also use the web interface at `http://www.ece.tufts.edu/ee/193HIP/provide.cgi`.

Starter code data

The data included with the starter code includes the XYZ matching functions for the CIE 2° 1931 observer.[1] These are in the variables `X`, `Y`, and `Z`, and the corresponding wavelengths for the samples are in `wavelength`.

Problem 1: Colorspace plots

1. We've looked at the XYZ spectral locus plot several times in the last couple weeks. Use MATLAB to generate this plot, using the data provided with the starter code. The `plot3` command can be used to plot a function in 3 dimensions. You don't need to worry about color; just plot the shape. *Hint: This is very easy if you know what you're trying to do.*
2. Plot the spectral locus in xy-chromaticity space. The shape will be familar when you have the right answer, but you may need to use `axis equal` or otherwise tweak the x and y axes.
3. Find the XYZ values of the RGB display primaries from HW 2, and plot the corresponding gamut on the chromaticity diagram. The MATLAB command `area` may be useful here.

Problem 2: Metacow done properly

Your metacow images from Homework 2 made one very bad approximation: after computing the L/M/S cone responses, we displayed them as if the L-cone mapped directly to a red pixel value, M to green, and S to blue. While the images looked roughly correct, the colors weren't right.

Now that you know about color spaces, you have the tools to do this correctly. Instead of computing cone responses, compute the cow image in XYZ color space.

Your computer monitor (probably) uses a standardized RGB color space known as sRGB, so to display the image properly you'll need to convert from XYZ to sRGB according to the following formula:

$$\begin{bmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix}$$

Finally, sRGB specifies a particular gamma curve, which you can read about on Wikipedia: `https://en.wikipedia.org/wiki/SRGB`. Apply the sRGB gamma to your data and then display it with `imshow`.

[1]There have been a few minor revisions since that time, but the 1931 definition is still in wide use. Data downloaded from `http://cvrl.ioo.ucl.ac.uk/index.htm`

Problem 3: Exploring YCbCr

YCbCr (or YCrCb) is part of the YUV colorspace family — a group of colorspaces with origins in TV broadcasting which split the image into luminance (brightness) and chrominance (color) components.

1. Load the image `cooking.png`.
2. Convert this image to YCbCr with MATLAB's `rgb2ycbcr`, and then blur the Y channel with a gaussian kernel (use `filter2`).

   ```
   [x, y] = meshgrid(-10:10); % Build square matrices with the specified range
   % Standard deviation of the blur; you may need to increase the meshgrid
   % range above if you make sigma significantly larger.
   sigma = 5;
   blur = exp(-(x.^2 + y.^2)/(2*sigma^2)); % Compute the gaussian blur
   blur = blur / sum(blur(:)); % Normalize to 1
   ```

3. Finally, convert back to RGB and display the image. Repeat this for the Cb and Cr channels.

Experiment with different values of `sigma` (which varies the radius of the blur). At approximately what value of `sigma` does the blur become evident for each channel?