# 7	ECE 253a	Digital Image Processing	Pamela Cosman	10/17/11

HOMEWORK 3

Due Wednesday October 26 in class

Comment on Matlab's imshow command: Depending on the version of Matlab, you can use imshow with type uint8, in which case the input should be integer-valued between 0 and 255, or you can use it with type double, in which case the input should be floating point between 0 and 1. If you have an array of type uint8 with integers between 0 and 255, and you convert it to double, and then immediately display with imshow without first rescaling the values to go from 0 to 1, the image will appear basically all white (since imshow is expecting values between 0 and 1, so everything from 1 to 255 will be interpreted as 1 = white). Likewise, if you have an array of type double with values between 0 and 1, and you convert it to uint8 and display with imshow, the image will appear all black (since all the values will be 0 or 1 which both look rather black when the scale is expected to go up to 255). So basically you can use either data type, you just need to rescale to make sure the range is right.

1. **Chromaticity diagrams**

   On the web site you will find a text file called `cie` which contains color matching functions in the XYZ coordinate system. The format of the file is `wavelength X Y Z` from 380 nanometers to 780 nanometers. To read it in:

   ```
   load cie -ascii
   ```

   (a) For this problem we ask you to produce hardcopy of your plots as well as of your matlab commands. On one graph, plot the color matching functions, $X(\lambda)$, $Y(\lambda)$, $Z(\lambda)$. On another graph, plot the xy chromaticity diagram. Connect the "line of purples" on your diagram.

   (b) The conversion from the CIE XYZ space to the NTSC receiver primary system R_N, G_N, B_N is given by the following linear transformation:

$$\begin{bmatrix} R_N \\ G_N \\ B_N \end{bmatrix} = \begin{bmatrix} 1.910 & -0.533 & -0.288 \\ -0.985 & 2.000 & -0.028 \\ 0.058 & -0.118 & 0.896 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

   The Society of Motion Picture and Television Engineers (SMPTE) made its own receiver primary color coordinate system. The conversion from the CIE XYZ space to the SMPTE receiver primary system R_S, G_S, B_S is given by the following linear transformation:

$$\begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} = \begin{bmatrix} 3.508 & -1.741 & -0.544 \\ -1.069 & 1.977 & 0.035 \\ 0.056 & -0.197 & 1.051 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Suppose there existed two sets of phosphors which exactly corresponded to the NTSC and SMPTE primaries. Inside the xy chromaticity diagram, plot the two triangles that correspond to the color gamuts of these two sets of phosphors. Does it appear that the NTSC or SMPTE primaries provide a larger gamut?

2. **Adding colors (Paper and pencil problem):**

   (a) Color C has tristimulus values T_1, T_2, and T_3. W denotes the reference white for the system. We form an additive mixture of colors: $A = \frac{1}{2}(C + W)$. What are the tristimulus values for A? What are the chromaticity coordinates for A?

   (b) The numerical values for the chromaticity coordinates for color C are $r = 0.1$ and $g = 0.1$. Where on the chromaticity diagram would color A be represented? Prove your result. (Include numerical values for the chromaticity coordinates for color A if that's possible from the information given).

3. **HSI processing for color images:**

   Read the image face.tif into Matlab and display it with imshow:

```
a = imread('face.tif','tif');
imshow(a)
```

   You can break it into three separate color planes and look at them as follows:

```
>> r = a(:,:,1);
>> imshow(r)
>> g = a(:,:,2);
>> imshow(g)
>> b = a(:,:,3);
>> imshow(b)
```

   You can convert the RGB planes to HSI and back again using the routines rgb2hsi.m and hsi2rgb.m which are based on the formulation from Gonzalez and Woods (except that we keep the angles in radians, not degrees).

```
[h,s,i] = rgb2hsi(r,g,b);
```

   The rgb2hsi routine expects inputs in the range of 0 to 255; check to be sure that your input ranges are correct. The h,s,i outputs from it are between 0 and 1.

   After processing in the HSI domain, you can convert back using hsi2rgb to obtain new color planes r1,g1,b1. For hsi2rgb.m, the h,s,i inputs must be between 0 and 1. The r,g,b outputs, however, are not necessarily restricted to the range 0–255. These 3 separate color planes of size 256 by 256 can be stuck together into a single color image array of size 256 by 256 by 3 using Matlab's built-in command reshape:

```
newim = reshape([r1 g1 b1], 256, 256, 3);
```

and then you would need to rescale it to look at it with imshow:

```
newim = newim/255;
imshow(newim)
```

(a) Histogram equalization on the I component:
Convert the RGB color planes to HSI. Use Matlab's histeq command to equalize the I component. Convert the H, S, and equalized I components back to RGB. After converting back to RGB (and before rescaling by 255 for display), you will find that some of the values exceed 255. Explain why this happens.

(b) The RGB values must be reduced to the usual 0 to 255 range. This can be done in several ways, including linearly rescaling all the values, and truncating the values. For each approach (linear rescaling and truncating) display and compare the original image and the histogram equalized version. Which approach provides the most contrast? What can you say about the saturation? Do you see any shifts in hue? Explain what you see.

(c) Equalize each of the original RGB color planes separately, each one getting equalized over its own histogram. Compare against the image that was equalized in HSI space. What does this separate equalization do to the hue and saturation?

(d) In the HSI space, try a few different things on the saturation plane that you think will increase the saturation. Convert back to RGB. Decide how to deal with the out-of-range problems, and compare the resulting image against the one which had the I plane processed. Are there areas which you successfully made more saturated? Did you pay a price for it in terms of changing intensity and/or hue?