ECE 253a	Digital Image Processing	Handout #8	Pamela Cosman	10/23/08

HOMEWORK 3

Due Friday October 31 by start of class

Comment on Matlab's imshow command: 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. **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?

2. **Chromaticity diagrams:**

On the web site, you will find a text file called `cie` which contains the 1931 two degree color matching functions in the XYZ coordinate system. (Aside: "Two degree" refers to the conditions for the color matching experiment. The size of the color splotches fills a circle that subtends two degrees of visual angle, so these curves are appropriate for foveal (in the fovea) vision. There are also 10 degree color matching functions relating to larger color patches.) The format of the file is `wavelength X Y Z` from 380 nanometers to 780 nanometers. The plain text file can be read into Matlab using `load cie -ascii`.

(a) On one graph, plot the color matching functions, $X(\lambda)$, $Y(\lambda)$, $Z(\lambda)$. Next, plot the xy chromaticity diagram. Connect the "line of purples" on your diagram. This is how color plate IV of the textbook is obtained. If you want grid lines on your plot, use the matlab command `grid`.

(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:

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

Convert the XYZ data to $R_N G_N B_N$ data. Plot the $R_N G_N B_N$ color matching functions. Just from looking at your plot, if you had a monitor with CRT phosphors that exactly corresponded to these NTSC typical ones, what range of monchromatic test lights would you be unable to match (display) on your monitor?

(c) Plot the rg chromaticity diagram. Connect the "line of purples" on your diagram. By hand, shade in the region that corresponds to the color gamut of this system: those colors which you can actually display on your monitor. What are the tristimulus values of the $R_N G_N B_N$ reference white in the $R_N G_N B_N$ system? Plot this point on your chromaticity diagram.

3. **Change of reference white**

This is a pencil-and-paper problem, no Matlab. We are given tristimulus values T_1, T_2, and T_3 for a color C. The tristimulus values are relative to a reference white W_1. In other words, for our color, the tristimulus values are given by

$$T_1(C) = \frac{A_1(C)}{A_1(W_1)}$$

$$T_2(C) = \frac{A_2(C)}{A_2(W_1)}$$

$$T_3(C) = \frac{A_3(C)}{A_3(W_1)}$$

You may interpret the A's as power knob settings in the color matching experiment. What would the new tristimulus values $\hat{T}_1$, $\hat{T}_2$, and $\hat{T}_3$ be for our color in a coordinate system that uses the same primaries but a reference white W_2? Derive your expression in terms of tristimulus values in the original coordinate system. (So your expression for the $\hat{T}$'s should involve only T's, no A's.)

Hint: What are the tristimulus values for the color W_2?