ECE 253a	Digital Image Processing	Handout #12	Pamela Cosman	11/3/08

Solutions for Homework 3

1. HSI processing for color images:

a) The conversion is done with rgb2hsi, and the processing can be done with histeq as follows:

```
[h,s,i] = rgb2hsi(r*255,g*255,b*255);
newi = histeq(i);
[newR,newG,newB] = hsi2rgb(h,s,newi);
```

We see that the new R,G,B planes have values substantially larger than 255. The largest one is in the red plane:

```
>> max(max(newr))
ans =
   492
```

The intensity of the image is computed using the following formula:

$$I = \frac{1}{3}(R + G + B)$$

where R, G and B are scaled between 0 and 1. Thus, the maximum intensity is 1 and is achieved only if $R = G = B = 1$. As discussed in lecture, suppose we had a region with color $R = 1$, $G = B = 0$. Then its intensity is 1/3. Suppose further that because of histogram equalization we would like to increase the intensity to 1/2. Increasing the intensity has the effect of multiplying the R, G and B values with the same factor. Thus, we need to multiply with 3/2. However, the R value, which was already at maximum, now becomes 3/2, which is greater than the maximum possible value of 1. Thus, histogram equalization in the HSI space can lead to RGB values outside the range.

b) Two simple choices are truncating and linear rescaling. For rescaling, two choices immediately come to mind. 1) In order to keep hue and saturation more-or-less unchanged, the R, G and B values should all be multiplied by the same factor. Scaling the image so that all color planes are between 0 and 255 clearly does this, provided we use the same scale factor for each of them, in which case some of the R,G,B planes will use less than the full 0–255 range. The rescaling makes the image look darker. This is most pronounced in the white regions of the image.

The rescaling might make it look as though some hues are changed. Consider the white paint stripe on the girl's face. The paint stripes become somewhat green after the histogram equalization and rescaling operation. Crop out a tiny portion of the image that has some of the lower white paint stripe in it:

a = face(100:128,100:128,:);

and look at the actual numbers in the array. The Red values tend to run around 200, the Green values there are 255, and the Blue values are 190. So this splotch of paint, although it looks white to us, actually has somewhat more green content in it.

After histogram equalization, the values there tend to run around [r,g,b] = [250,290,220]. If one looks at this image directly with imshow, the 290 value gets truncated to 255, so the color looks white.

After rescaling all three color planes by dividing by 492 and then multiplying by 255 to maintain the 0-255 range, that splotch has values around [r,g,b] = [129,150,114] which looks somewhat greenish. I guess at lower intensity levels, the fact that that area is slightly more green is more noticeable, whereas when all three color planes are brighter, it just looks white.

2) The other rescaling choice is to rescale each plane separately. Then each one will use its full range 0–255, and the image will be brighter on average. However, the hues will shift.

Truncating will leave those values alone that are between 0 and 255. Thus, in regions where no truncation occured, the image looks brighter and is enhanced by the equalization. However, where color values were truncated, the hue and saturation are changed (unless miraculously the truncation was identical in all three planes for a given pixel). Overall, the image looks brighter compared to the linear rescaling options, since a lot of the color coordinates are at their maximum. The white background appears unchanged since it has maximum intensity and histogram equalization does not change the minimum and maximum intensities.

c) When we do equalization on each of the color planes separately, the ratios between the R, G and B values are changed and some of the colors look different, that is, the hue and saturation are now different in some places from the original image.

d) Here there are lots of acceptable answers. Taking the S matrix (the saturation plane) and multiplying by 2 will cause some values to exceed 1, so you can threshold the values at 1. Transforming back to r,g,b shows that the new r,g,b values have even higher maximum values than before. Using the thresholding method to deal with these out-of-range values will show that this tweak on the saturation has not produced a dramatic effect. The face looks a little redder. One place that the saturation has noticeably increased is for the blue paint stripe, which is a “richer” blue than before.

There are ways to get more dramatic effects of increasing saturation. For example, you can threshold the S matrix below at 0.2, that is, any value of saturation less than 0.2 is remapped to 0.2, and all other values are left alone. Various patches in the image that were previously white will now have noticeable color effects (the white paint stripe becomes slightly yellow, and the background and the white of the eye become tinged light blue).

Doing the same thing with a minimum saturation value of 0.5 instead of 0.2 cause the resulting image to look very strange (e.g., the white of the eye and the background white now look very blue). In effect, what this processing does is to take those colors which have very low saturation, which are more or less white, and force them to declare themselves as being one color or the other. So a white light which has just a bit more blue than it has other wavelengths gets forced to become very blue.

It is definitely possible to increase the saturation in ways that make the image look appropriately richer and more vivid, but indiscriminate tweaking of the S plane is more likely to throw a lot of colors out of whack.

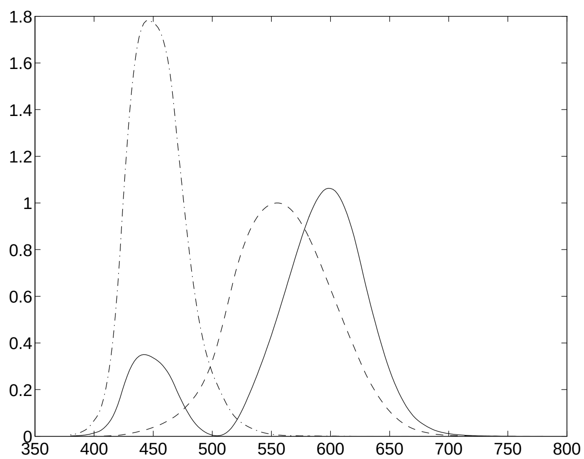
2. CHROMATICITY DIAGRAMS

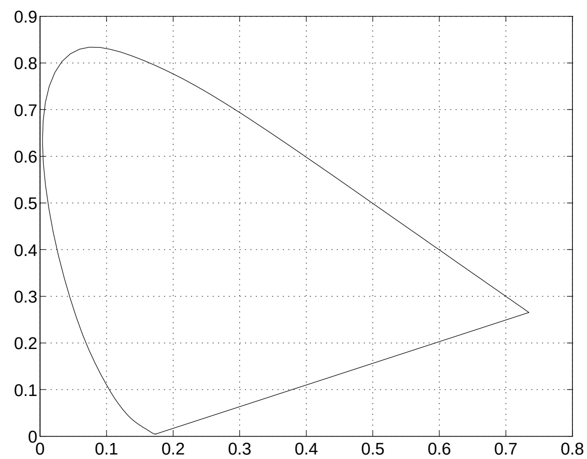
(a) The following steps will load the color matching functions of the XYZ coordinate system and display them on the screen:

```
load cie -ascii
l = cie(:,1);
X = cie(:,2);
Y = cie(:,3);
Z = cie(:,4);
plot(l,X,'r-',l,Y,'g--',l,Z,'b-.');
(See plot below left)
```

For the chromaticity diagram we have to normalize and connect the line of purples. By adding the first entry of `x` and `y` to the end, the plot will be connected.

```
x = X./(X+Y+Z);
y = Y./(X+Y+Z);
x = [x;x(1)];
y = [y;y(1)];
plot(x,y);
grid;
(See plot below right)
```





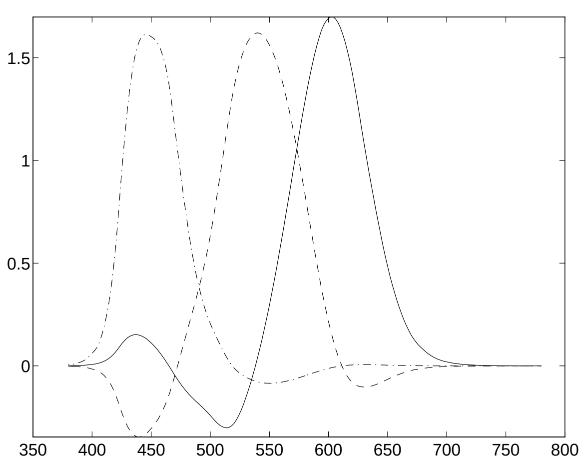
(b) For the conversion from the XYZ space to the $R_N G_N B_N$ space:

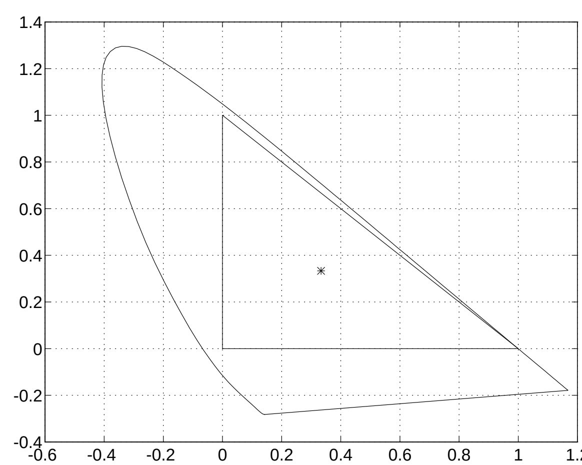
```
rgb = cie(:,2:4) * [1.910 -0.533 -0.288;-0.985 2 -0.028;0.058 -0.118 0.896]';
R = rgb(:,1);
G = rgb(:,2);
B = rgb(:,3);
plot(l,R,'r-',l,G,'g--',l,B,'b-.');
```

As we can see from the plot (below, left), at every monochromatic light there is at least one matching function which is negative. However, since the monitor phosphors can not emit negative intensity, this implies that no monochromatic light can be matched. Another way of checking this is that, the color gamut triangle is entirely contained inside the chromaticity "horseshoe"– it never quite includes the border.

(c) For the chromaticity diagram we have to normalize again and connect the line of purples. The displayable colors can then be found in the triangle defined by the three points $(0, 0)$, $(0, 1)$ and $(1, 0)$. Since the tristimulus values are normalized to a reference white, the color coordinates of the $R_N G_N B_N$ reference white in the $R_N G_N B_N$ system must be $(1, 1, 1)$. Thus, in the chromaticity diagram the reference white can be found at $(1/3, 1/3)$.

```
r=R./(R+G+B);
g=G./(R+G+B);
r=[r;r(1)];
g=[g;g(1)];
plot(r,g,'-',[0 1 0 0],[0 0 1 0],'-',1/3,1/3,'*');
grid;
(See plot below right)
```





Left: Color matching functions for $R_N G_N B_N$ color system
Right: Chromaticity diagram for $R_N G_N B_N$ color system

3. Change of reference white

   Tristimulus values for a color C are the ratio between the "power knob settings" used to match the color C in the color matching experiment, and the "power knob settings" that were used to match the reference white for that coordinate system.

   For the tristimulus values in a coordinate system that uses reference white W_2, we simply put, in the denominator of each ratio, the "power knob settings" that were used to match W_2.

$$\hat{T}_1(C) = \frac{A_1(C)}{A_1(W_2)} = \frac{A_1(C)}{A_1(W_1)} \times \frac{A_1(W_1)}{A_1(W_2)} = \frac{T_1(C)}{T_1(W_2)}$$

$$\hat{T}_2(C) = \frac{A_2(C)}{A_2(W_2)} = \frac{A_2(C)}{A_2(W_1)} \times \frac{A_2(W_1)}{A_2(W_2)} = \frac{T_2(C)}{T_2(W_2)}$$

$$\hat{T}_3(C) = \frac{A_3(C)}{A_3(W_2)} = \frac{A_3(C)}{A_3(W_1)} \times \frac{A_3(W_1)}{A_3(W_2)} = \frac{T_3(C)}{T_3(W_2)}$$