ENGG 111 - Homework Assignment 1

Preliminaries:

First assignment should be easy. These exercises are quite simple, they are devised to ensure that you understand the various types of images, and have some understanding of intensity transformations.

Submit homework assignments by email, one file for the assignment (First_Last_HW1.pdf) and one file for all the appendix files (images, code, whatever else in: First_Last_HW1.zip).

HW1 is due before next Friday's class (4/10/15).

Part 1 - Matlab Image Representation - Gray Scale, Look Up Tables

A - Image Representation

Using the stock image provided by Matlab "AT3_1m4_01.tif" determine the type of file it is by using the command "imfinfo", what is it? *(NOTE: the character before the "m" in the file name is a one (1).)*

Load the image into a variable, "im" for example. What is the data type of the image and what structure is used to store (single plane or grayscale, single plane with LUT, etc.). This is revealed by using the "who" or "whos" commands.

Convert explicitly the image to type double. What do you need to do to ensure that it conforms to Matlab's convention for images of type double.

Show a panel with multiple images (figure + subplot for example)

a) the original image unaltered shown using imshow() and imagesc()

b) the double image

c) the double image after proper correction

give titles to every subplots showing what they are.

Write a script that does all the steps above and submit it with your homework.

B - Grayscale

Write a short Matlab function which given a grayscale image will return the same figure as an indexed image: You should be able to type:

[imIndex, imMap]=gray2index(im);

You should remember that the imIndex is an array of indexes (integer) pointing into the imMap array. The imMap array should contain values ranging from 0.0 to 1.0, for each of the R G and B components, even though we have a gray scale image. The correct arrangement for the two arrays should be imIndex(1:N,1:M) for an image of N rows by M columns, with the imMap(1:C,1:3), C being the number of gray values needing to be represented. You can use the function imhist to get an idea of what values are in the image.

Do not use the Matlab function "gray2ind", that would be too easy, I want to see that you know how the data is organized in the two image types, and that you have some proficiency with Matlab programming. Make sure to use vectorization as much as possible to make your function run efficiently. Given what you know about the data type of the original image, what difference is there between the original image "im" and the index table "imIndex" ?

Now using what you know about creating indexed images from grayscale images, modify your code as follows:

[imIndex, imMap]=gray2index(im,n);

Modify it so that it takes either one or two arguments (see documentation on "variable number of arguments" in Matlab). Now, instead of creating a table of gray values that has length C, the number of gray values in the original image, you should return a map that has n entries. Remember that this will also affect your imIndex array contents.

Show a panel based on "AT3_1m4_01.tif" which shows it unmodified (as many levels as in the original image), 64 levels, 16 levels, and 4 levels, as I showed in class, but using your own code. Show titles indicating the number of gray values. Note: I expect the values table (map) to be of length n. In class I cheated by duplicating values in the table.

Part 2 - Exploring histogram equalization

C - Histogram equalization on a color image.

I indicated in class that histogram equalization applies only to intensity (grayscale) images. However, there may be ways to get around that.

As an exercise, use as an example the stock image "peppers.png" and apply histogram equalization to the individual planes as it has been suggested in class. Can you comment on the results, comparing to the original image. Particularly, comment on the resulting shades of color and the visibility of details.

You are to devise a way to do a better job of this. Here is a suggested method:

a) compute an intensity image corresponding to the original one

b) perform equalization on that image

c) scale the intensity of the original color image (all three planes) with the ratio of the new and old intensity images

what does it look like?

Note, that while the image will not be perfect, you should see an image in which the tones are much more similar to the original images, and some of the details much more visible than in the original image. Using this method (which is imperfect you will see why later) should work better than the 3 plane equalization.