Image Processing I

EECE/CS 253 Fall 2011

Assignment 3: Color Correction of Images

Due at 12:00 midnight on Monday 10 October 2011

The goals of this lab are to learn how to (1) apply point operations to the R, G, and B bands of color images to alter their colors, (2) convert images to HSV space, perform operations on those bands, and convert the results back to RGB, and (3) generate a linear color correction matrix from corresponding pixels in two images.

In this lab assignment (and all the others) you will be writing your own image processing functions. You may not use the functions from the `matlab` image processing toolkit by themselves or within your functions to solve the problems in the labs. You may compare the results of your own functions against those of the IP toolkit for your own information but those results may not be used in your lab reports.

Help can be gotten on any `matlab` command (or function) by typing `help command_name`, *e.g.* to get help on `imwrite()`, type `help imwrite` at the command prompt. Another way is to press the <F1> key, select the **index** tab in the window, and type the command name into the search box.

In completing this assignment (and the others) if specific images are not supplied, you may use any images you like providing that they are of the type specified by the problem description (*e.g.* 24-bit truecolor). Please do not use images that are obscene or gruesome. In general, an image is OK if you could show it to your grandmother without upsetting her or embarrassing yourself. A wide variety of images are available under creative commons license on the web site `www.flickr.com`. If you use one or more of these in your report, be sure to credit the photographer.

1. This problem explores the changes in color that occur when the individual bands of a true-color image are transformed independently of each other.

   Find a truecolor image that contains a wide range of colors.

   (a) Use the gamma correction function you wrote for Lab Assignment 2 to increase and decrease the overall brightness of each band separately. Display the original image and the 6 altered versions. Describe the differences between the images.

   (b) Find the mean intensity of each band. Threshold each band. That is let

   ```
   J = uint8(zeros(size(I)));
   J(:,:,1) = uint8(255*(I(:,:,1)>mr));
   J(:,:,2) = uint8(255*(I(:,:,2)>mg));
   J(:,:,3) = uint8(255*(I(:,:,3)>mb));
   ```

   where `I` is the original image and `mr`, `mg`, and `mb` are the mean red, green, and blue intensities.

Display J. How many colors do you see? What are they? Explain why this particular set of colors is present. What are the colors that could possibly be present in an image created this way?

2. This problem asks you to write color conversion programs and to perform color manipulation in the HSV color space.

   Write a program, `RGBtoHSV` to convert an image from RGB to HSV format. The syntax should be:

   `[H,S,V] = RGBtoHSV(R,G,B);`

   where R, G, and B are the red, green, blue bands from an image, I. H, S, and V are of class `double`. Write the program so that H ranges from 0 to 2π with $0 =$ red, $\frac{2}{3}\pi$ $=$ green, $\frac{4}{3}\pi =$ blue, and $2\pi =$ red. S ranges from 0 to 1 with $1 =$ full saturation. V ranges from 0 to 255 with non-integer values likely.

   After operating on the H, S, and V bands as described below, you will need to convert the results back into R, G, and B bands of type `uint8`. For this you will need to write another program, `HSVtoRGB`,

   `[R,G,B] = HSVtoRGB(H,S,V);`.

   H and S, must be of class `double` and lie within the ranges described above. V may be either class `double` or `uint8`. In either case, the values of V must lie in the range $[0, 255]$. R, G, and B should be returned as type `uint8`.

   For conversion of RGB to HSV, use the definition given in

   `EECE253_05_ColorCorrection.ppt`, pp. 78-82.

   For conversion of HSV to RGB, use the definition given in

   `EECE253_05_ColorCorrection.ppt`, pp. 83-91.

   Those are the exact definitions; use them in your program. There are a number of similar, faster somewhat different approximations listed in the literature, *e.g.* , the `RGBtoHSI` algorithm listed on p. 299 of Gonzalez and Woods and pp. 211-214 of Gonzalez, Woods, and Eddins. But don't use those. Start with the definitions in the presentation and write your own algorithms to implement them. Comment your code liberally and include it in the appendix of your report. Use your programs in problem 2 of the lab assignment.

   The three bands must be recombined into a single image. If R, G, and B were originally extracted from a three-band color image I, then the following Matlab code will recombine the bands:

   `J = reshape([R G B],size(I));`

   (a) Use the same image you used in problem 1. Apply your gamma function to the V band, both to increase it and to decrease it. Reconstruct two new images using the altered V bands via your `HSVtoRGB` program. Describe the results.

(b) *Linearly* increase and *linearly* decrease the saturation. Reconstruct two new images using the altered S bands via your program `HSVtoRGB`. Describe the results. Note that a linear increase may increase the value of the saturation of some pixels to > 1. What can you do to correct for that?

(c) Apply a constant circular shift to the H band to shift the colors in the green direction from hue $= 0$. Note that a circular shift of amount δ away from red toward green could cause the hues of some pixels to exceed 2π. These must be "wrapped around". That is, if $H(r, c) > 2\pi$ then replace $H(r, c)$ with $H(r, c) - 2\pi$,

Repeat the same experiment but shift the colors in the blue direction. Note that you must perform an analogous wrapping to deal with the pixels whose hues are shifted to below zero in value.

Reconstruct 2 new images using the altered H bands via your program `HSVtoRGB`. Describe the results. How do these results differ from those of problem 1?

3. Correction of discolored photos through the linear transformation of color vectors.

(a) Download the images,

`lance_armstrong_orig.jpg` and

`lance_armstrong_alt.jpg`,

from the lab assignment 3 folder on Blackboard. The first image is the original and the second is a color-manipulated version of the original. Select 15 (or more) different colors from the original image. That is, record the rgb values from at least 15 different pixel locations in the image. Select a wide range of colors and intensities. Ideally, you would select dark, medium, and bright pixels for each of the 6 primary colors (red, yellow, green, cyan, blue, magenta) and grayscale for a total of 21 colors from 21 pixel locations. But, of course, most images (including this one) do not contain all colors. I suggest that you use the Matlab function, `ginput`, to select locations from the image.

Select the same pixels from the altered image.

Make a table that lists the pixel locations and the corresponding rgb triplets from both images.

(b) Construct a linear transformation matrix from the selected colors[1] List the transform matrix.

(c) Transform the altered image using the transformation matrix. Display the original and the result, plot their pdfs, and comment on the differences between all three images.

[1] See pp. 110-145 of `EECE253_05_ColorCorrection.ppt`

4. Additional problem for graduate students. Please download and read the paper in

   `2010.FALL.ENG.EECE.253.01 Image Processing > Course Content >`
   `Other Materials > Miscellaneous`

   that I have labeled, "An excellent review of color spaces by Danny Pascale," and has filename

   `Pascale - A review of RGB color spaces.pdf`.

   You may limit your reading of it to the parts that pertain to this problem.

   Write a program that accepts an arbitrary RGB image I, a 3×3 RGB to XYZ matrix M, and an associated 3×1 white illuminant vector $\mathbf{v}$ (see p. 21). It should convert the image into X, Y, and Z bands and output them. *E.g.*

   `[X,Y,Z] = RGBtoXYZ( M,v,I );`

   M, v, X, Y, and Z are of type double. I is of type uint8. Also write a program that computes the inverse,

   `J = XYZtoRGB( M,v,X,Y,Z );`

   where J is uint8.

   Get an M and $\mathbf{v}$ matrix and vector from the table on p. 21 of Pascale's paper. Verify that the programs are inverses by using a $9 \times 9 \times 3$ block from an RGB image.

   Write two programs, one that converts X, Y, and Z into L^*, a^*, and b^* bands, the other that computes the inverse.

   `[L,a,b] = XYZtoLab( X,Y,Z );`

   `[X,Y,Z] = LabtoXYZ( L,a,b );`

   Verify that these two programs are inverses by using the same $9 \times 9 \times 3$ block from the X, Y, and Z bands that you generated from your RGB image.

   Determine the natures of the L^*, a^*, and b^* bands by performing some experiments as follows:

   Find a richly colored RGB image and convert it into L^*, a^*, and b^* bands using your programs, `RGBtoXYZ` and `XYZtoLab`. Indicate which M and v you used, and why you selected them. Then perform some point operations on the L band. Reconvert the resultant L^*s, and the original a^*, and b^* bands into RGB images using `LabtoXYZ` and `XYZtoRGB`. Do other (or the same) point operations on the a^* band alone, and an others on the b^* band alone and reconstruct the results. The goal is to determine the information carried in each of these bands. Use the original image and a subset (at least one for each band) of your modified RGB images to support your explanations.

In your report, group on the same page the images or pdfs that are to be compared so that it is easy to look from one to the other without having to jump around in the document.

Rules for laboratory assignments

1. Perform all the tasks listed in the instructions.
2. Explain the tasks you performed in detail.
3. Answer in writing in your report all the questions asked in the instructions.
4. Include in the report the original images you used and those resultant images that were specified in the instructions.
5. Include all computer code that you wrote and used, clearly documented, in an appendix.
6. All work must be yours and yours alone. Collaboration on the laboratory assignments is forbidden, with the following exceptions:
   (a) You may obtain help on any aspect of the homework from either Prof. Peters or the TA for this course.
   (b) You may obtain technical help on `matlab` from anyone you wish. However *you may not get direct help on the implementation of the specific algorithm* from another person except as noted in (a) above.
   (c) You may get help in obtaining the *input* images for the assignments from anyone you wish.
   (d) You may get help in the formatting or storing or transmission of your reports, but *not the content*, from anyone you wish.
7. Write your results in a clear laboratory report format using MS Word, WordPerfect, LaTeX, or any other word processor with which you can embed images in text. I prefer that the reports be submitted in .pdf format, but that is not required. Submit your report to me as a file on Blackboard. If for some reason this does not work, you may submit your report on a CD-ROM.
8. Assignments are due at midnight on the day specified in the instructions or in class. The grade on a laboratory report will be reduced by 10 points (out of 100) for every day (24 hours) that it is late.