EECE 253 IMAGE PROCESSING

LABORATORY ASSIGNMENT 2

Jack Minardi
9 September, 2011

Abstract

This paper reports the results of experiments done to explore point processing on images using Matlab. Point processing is the set of operations that change individual pixels in an image. Each pixel is changed independantly of other pixels in the image. In this report a few different techniques were utilized, such as brightness, contrast, gamma, lookup tables transformations, histogram equalization, and histogram matching. To note the results of these transformations, a histogram plot was used, that showed a histogram for R, G, B, and luminance bands of the image. Different results had different signatures in the histogram plots.

Table of Contents

Introduction

Point processing is one of the most common operations on images. It can be used to adjust the overall image as a whole, and is therefore used to change attributes such as brightness and contrast. This is done by adding or subtracting a constant from every value, or changing each value depending on its specific brightness. Both of these operations can be done through a lookup table (LUT) which maps each of the values between 0 and 255 to another set of values between 0 and 255. LUTs can be used to enhance the contrast in specific regions of the image.

This lab also explored means of adjusting an images qualities to more match another image. This can be accomplished by matching the histogram of one image to another. This has the effect of adjusting the colors and brightness to resemble as closely as possible the other image. Many programs were written to accomplish the tasks in this lab, and they can all be seen in the appendix of this paper. Throughout this paper histograms and probability density functions are used to examine the results of certain operations. These two graphs brake the image down into components so as to see the effects.

Description of Experiments

There were four experiments performed in this lab. Matlab was used for all the computation. All the scripts that were written can be found in the appendix and in the zip file this report came in. The results were somewhat subjective and discussions can be found in the report below.

Experiment 1. Image histograms and probability functions.

(a) Write a Matlab function to compute the histogram of a one-band image. Find and load a 24-bit truecolor image, preferably one with strong primary colors. Compute the R, G, B, and K

histograms of the image. The K histogram is the histogram of the value image (V = 1 3 [R + G + B]) or the luminance image (L = 0.2990R + 0.5870G + 0.1140B). Use one of the two; your choice. Since your histogram function operates on one-band images you will need to run it separately on all four bands. Plot all four histograms on the same graph so that the red band's histogram is plotted in red, the green band's in green, the blue band's in blue, and the value or luminosity histogram in black.

Some images will have a dominant color – often with one or more intensities equal to 255. That can cause one bin to be very large with respect to the others. When you plot such a histogram, the detail in the other bins becomes lost in the figure. If you find that to be the case with your image, use the command axis to scale the plot window so that you can see see the majority of the histogram clearly. For example if one bin has a count of 90000 and the next smaller count is 19000, the following command – executed before the print command – would make the graph much more readable: >> axis([1 256 0 20000]);

(b) Write a Matlab function to compute the probability density function of a oneband image. Have the pdf function use the histogram function that you wrote for part (a). Generate and plot the pdf for the R, G, and B, bands and K image just like you did in part (a).

(c) Write a Matlab function to compute the cumulative distribution function of a one-band image. Have the CDF function use the pdf function that you wrote for part (b). Generate and plot the CDFs for the R, G, and B, bands and K image just like you did in parts (a) and (b).

Experiment 2. Brightness, Contrast, and Gamma.

(a) Write individual Matlab functions for brightness, contrast, and gamma adjustment of images of class uint8. Each function should determine internally if the input image is one-band or multiband and treat it accordingly. The contrast function should generate a lookup table that is used to transform the bands of the input image. The gamma function should do the same. The brightness function may or may not use a LUT – your choice. Each function should accept an image matrix as input and output an image matrix of the the same size, number of bands and data class as the input. The brightness function should have one argument in addition to the image: a percentage, p. Compute actual brightness shift value, g, as p percent of the mean intensity value of the image. The contrast function should have two numerical arguments, the slope and the midpoint of the remapping line. The gamma function should have one argument, the gamma value such that $\gamma = 1$ leaves the image unchanged, $\gamma > 1$ brightens and $0 < \gamma < 1$ darkens the image.

(b) Modify the truecolor image you used in part 1 with each of the three functions. Create two

example outputs for each function, one with increased attributes (i.e., brightness, contrast, or gamma) and one with decreased attributes. The result will be 6 example transformations of the input image.

(c) For each of the six output images plot the R, G, B and K histograms as you did for the original image in part 1(a).

(d) Additional problem for graduate credit. Separate the original image into its three color bands – one single-band image per color. Perform experiments to determine the effects of applying the same point op, but with different parameter values, to each of the three single-band images. Recombine the images into one truecolor image and describe the results. For example, try using different contrast slopes and/or midpoints on the R, G, and B images, recombine the 3 bands back into a single truecolor image and report the results. From your experiments derive some general rules for the differential band behavior of these operators. For example: When $\gamma < 1$ is applied to the R band of the image, the resultant image looks . . . . This need not be exhaustive but perform enough experiments that you can predict what will happen when some combination specific combination of operators is applied the different bands of the same image. For example what would happen if we decrease the gamma of the blue band while increasing the gammas of the red and green band. List the experiments you performed and the results for each. Include three transformed images (your choice of transformations) that illustrate some of your conclusions. Include histogram plots with each of the three and relate the transformed histograms to what you see in the images.

Experiment 3. Piecewise linear remapping using a lookup table

(a) From the K CDF of the truecolor image that you used in the previous parts, identify the 33rd and 66th percentile gray levels. That is, find g where C DF (g + 1) < 1/3 and C DF (g + 2) ≥ 1/ 3 and k where C DF (k + 1) < 2/3 and C DF (k + 2) ≥ 2/3. 30 255 255 g33 g66 Look Up Table output input Figure 1: Plot of a look-up table of the type to be used in problem 3 . Then g + 2 is the 33rd percentile intensity level and k + 2 is the 67th percentile intensity. You will use these values to generate piece-wise linear lookup tables (LUT) below. Each LUT has 256 elements and maps values in the interval [0, 255] into the interval [0, 255]. Since Matlab indexes from 1, LUT(x+1) maps intensity x to a corresponding intensity, y. Each LUT should i. consist of 3 connected line segments, ii. should map 0 to 0 and 255 to 255, iii. have different slopes over the intervals [0, g], [g,k], and [k, 255], such that the differences, iv. LUT(2)-LUT(1) and LUT(g+1)-LUT(g) both equal the slope of the first interval, LUT(g+2)-LUT(g+1) and LUT(k+1)-LUT(k) both equal the slope of the second interval, and LUT(k+2)-LUT(k+1) and LUT(256)-LUT(255) both equal the slope of the third interval. See Figure 1 for a example. Note that Matlab's indexing means that the first interval, [0, g], is indexed by [1,g+1].

(b) Generate a LUT to contrast enhance the lower third of the gray values while diminishing the contrast of the upper two thirds.

(c) Generate a LUT to contrast enhance the middle third of the gray values while diminishing the contrast of the outer two thirds.

(d) Generate a LUT to contrast enhance the upper third of the gray values while diminishing the contrast of the lower two thirds.

(e) Plot the three remapping functions (the LUTs).

(f) Include copies of the remapped images. Visually inspect, describe, and compare the results.

(g) Plot the K pdfs for each of the remapped images with the critical points labeled. Note: The LUT is derived from the value or luminance image. But apply it to each of the three color bands individually and recombine the bands into a single truecolor image to examine the results.

Experiment 4. Histogram equalization and matching.

(a) Write a program to histogram equalize a truecolor image.

(b) Use the program on a truecolor image.

(c) Include the images and describe the results.

(d) Plot the pdfs of the result (as in Problem 1).

(e) Write a program to generate a LUT that matches the histogram of one image to that of another. It will be easiest to write the program to accept the pdfs from each image and to generate the LUTs from those. Recall a LUT is generated from the CDFs of the images – not the pdfs – but the CDFs are generated from the pdfs.

(f) Select two truecolor images that have noticeably different color characteristics. Note that they need not be the same size. Remap the first image so that its luminosity (or value) histogram matches that of the second as closely as possible. Display the resulting image and plot the pdfs of the result (as in Problem 1). Comment on the appearance of the remapped image relative to the original and to the other image.

Results of Experiments

The image used throughout this report is shown below in Figure 1.

Figure 1. Original image. Source: http://www.imagegossips.com/2011/01/nature-beauty/

Experiment 1

a) The Matlab function written to compute the histogram is called histo.m and the function used to plot all bands is called fullHist.m. Both functions can be found in the appendix. Below is the histogram of Figure 2.

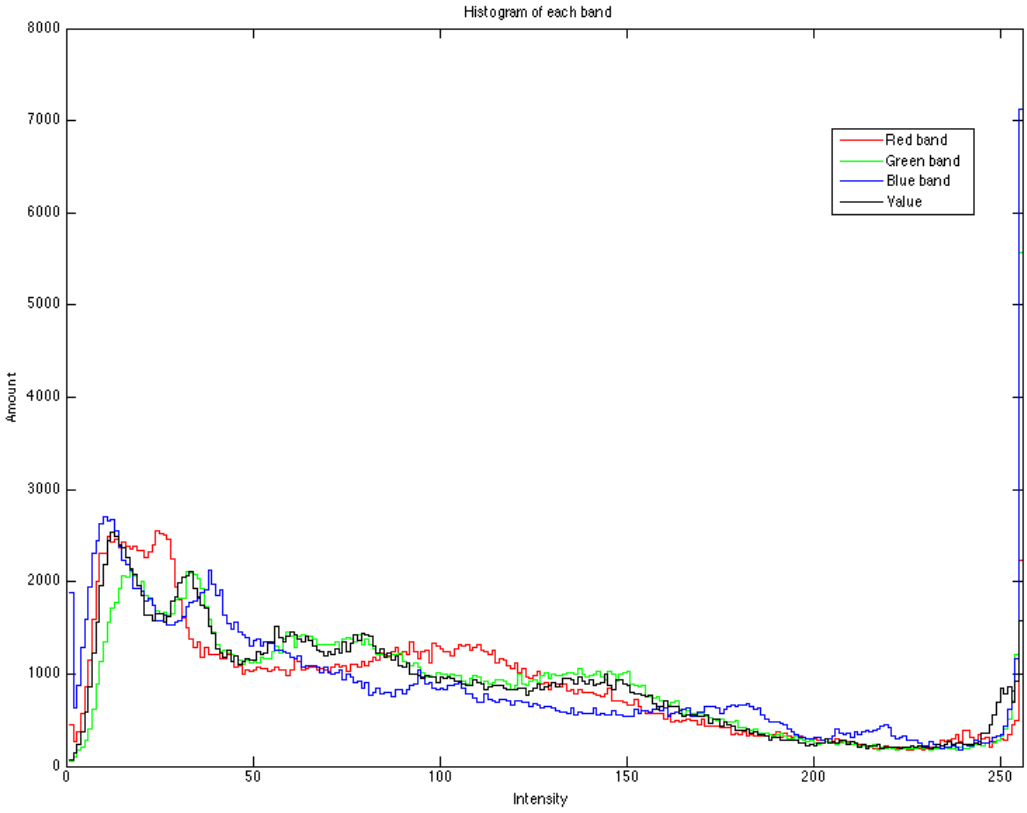

Figure 2. 3 bands and intensity histogram of Figure 1.

b) The program written to compute the probability density function is called probo.m. Below is the PDF of Figure 1 on all bands

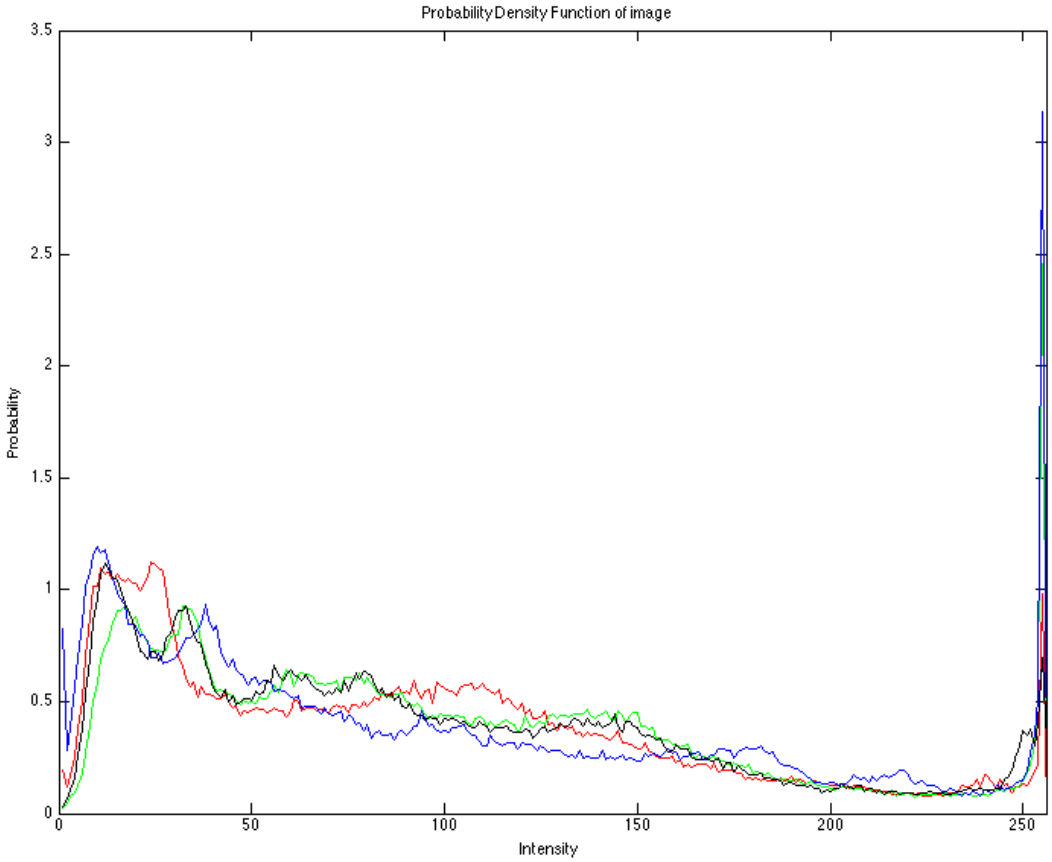

Figure 3. PDF of Figure 1

c) The program written to compute the CDF of an image is called cdfo.m. Below you can see its output from Figure 1.

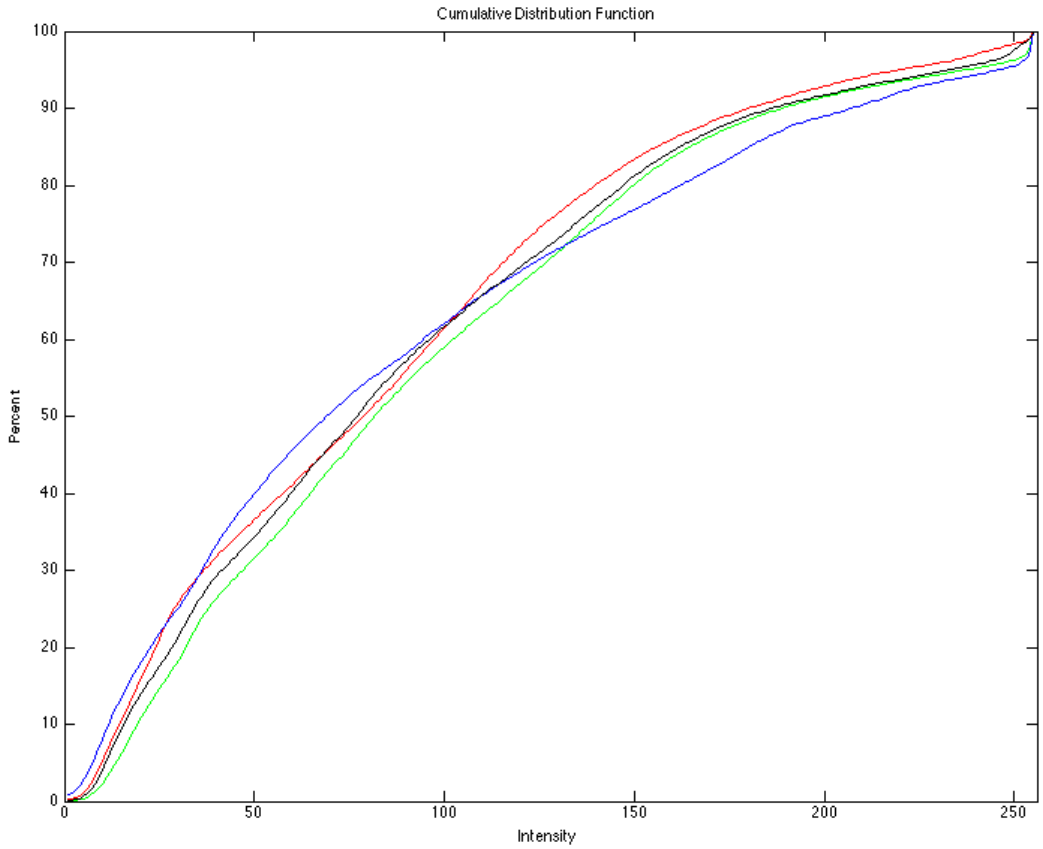

Figure 4. CDF of Figure 1.

Experiment 2

a) The following functions were written for this part:
brightness.m - adjusts the brightness of a 1-band image
contrast.m - adjusts the contrast of a 1-band image
gamma2.m - adjusts the gamma of a 1-band image

The programs can be found in the appendix.

b) and c) The following 6 images are the original true color images with increased and decreased values from each of the functions. They are labeled accordingly.

Figure 5. Brightness decreased

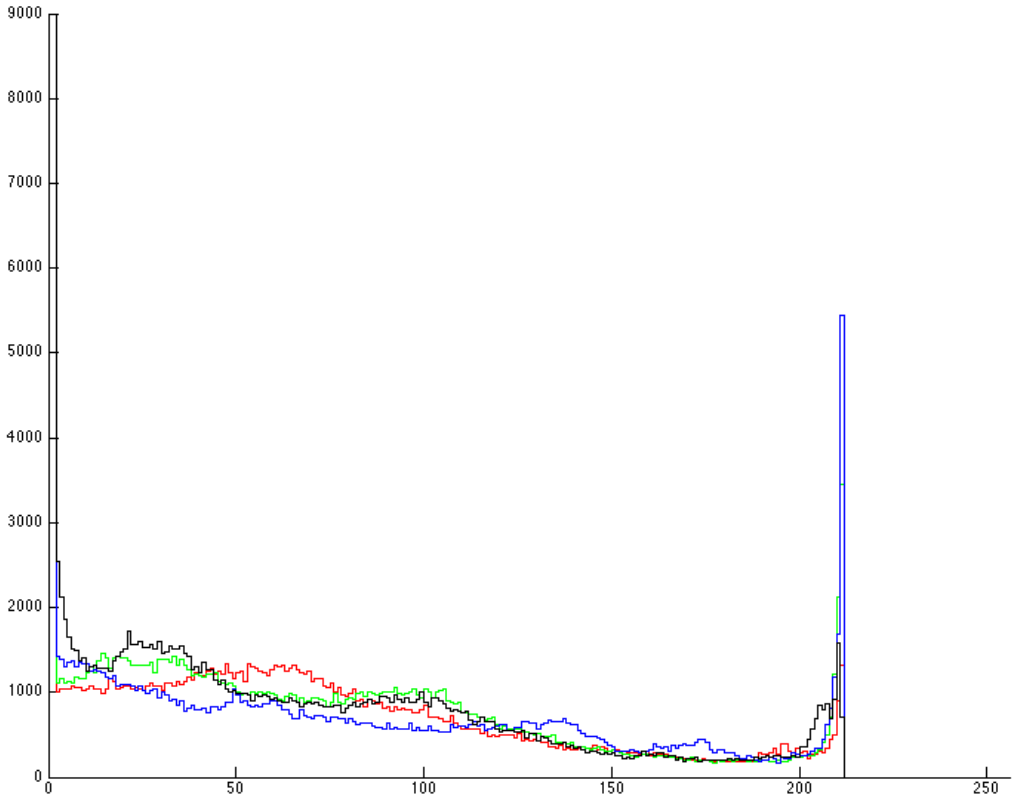

Figure 6. Histogram of Figure 5

Figure 7. Brightness increased

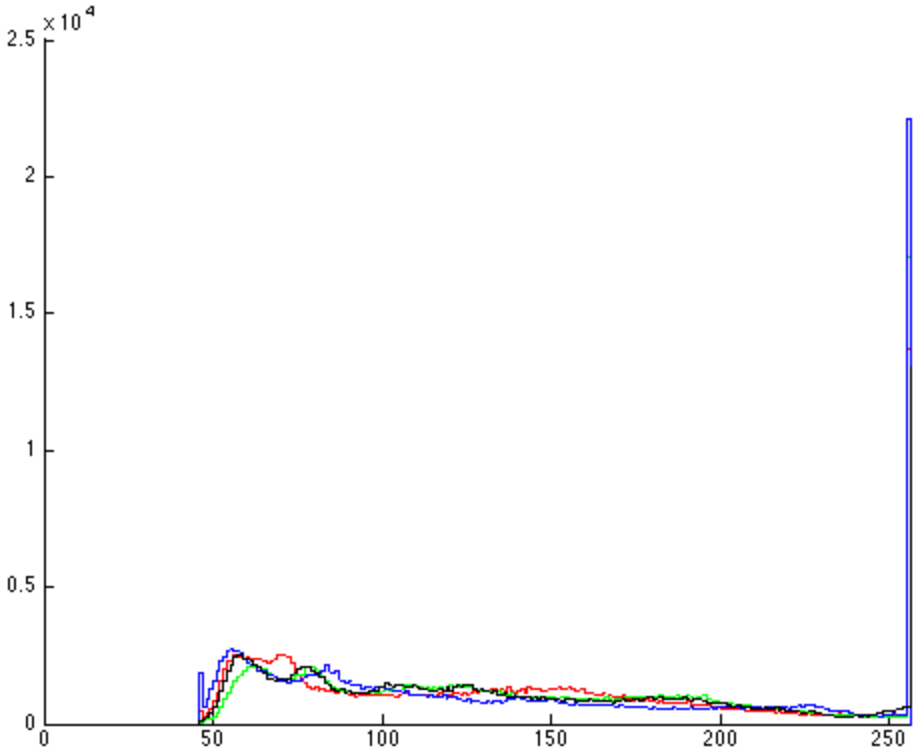

Figure 8. Histogram of Figure 7

Figure 9. Contrast decreased

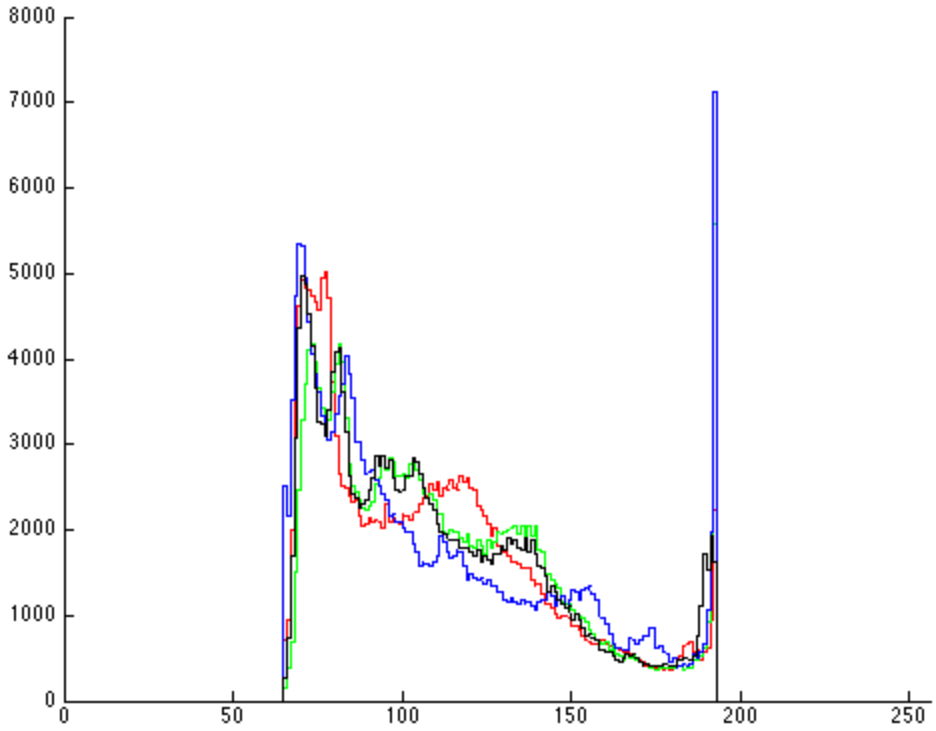

Figure 10. Histogram of Figure 9

Figure 11. Contrast increased

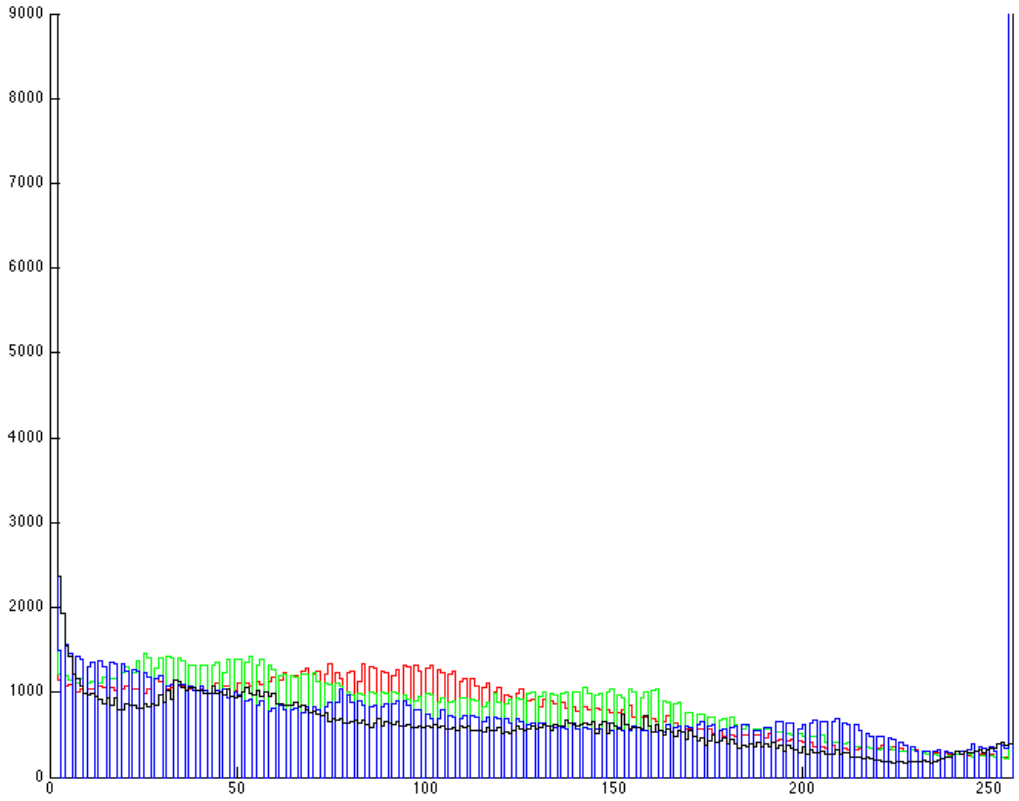

Figure 12. Histogram of Figure 11

Figure 13. Gamma decreased

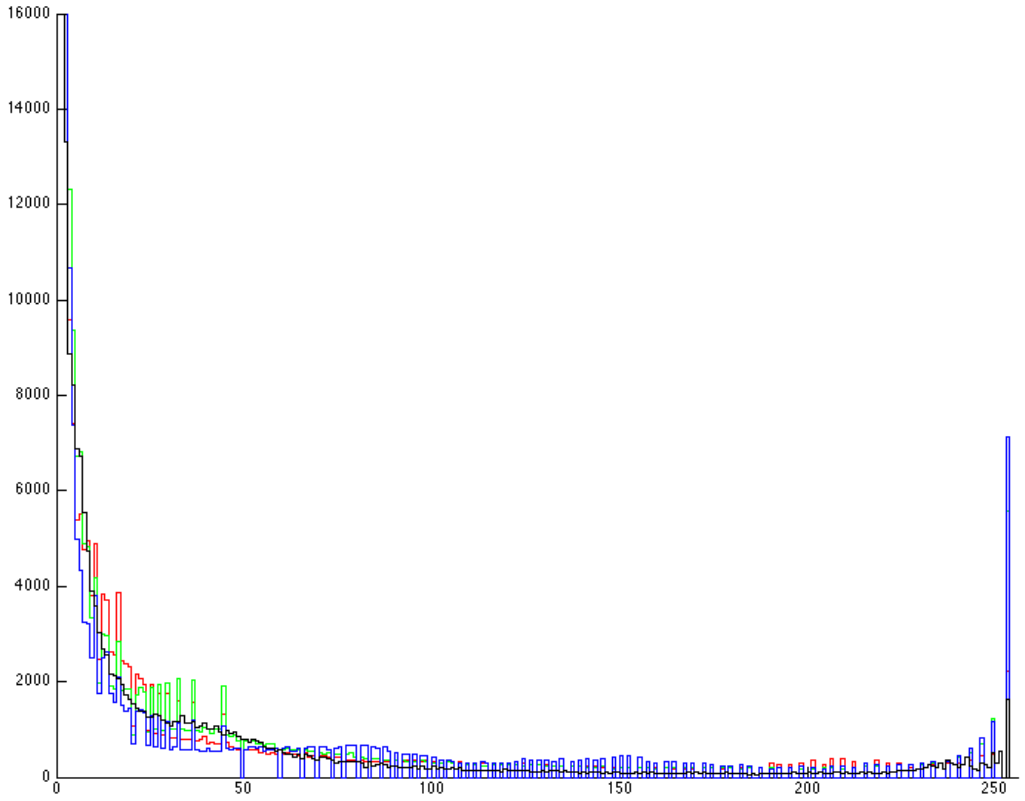

Figure 14. Histogram of Figure 13.

Figure 15. Gamma increased

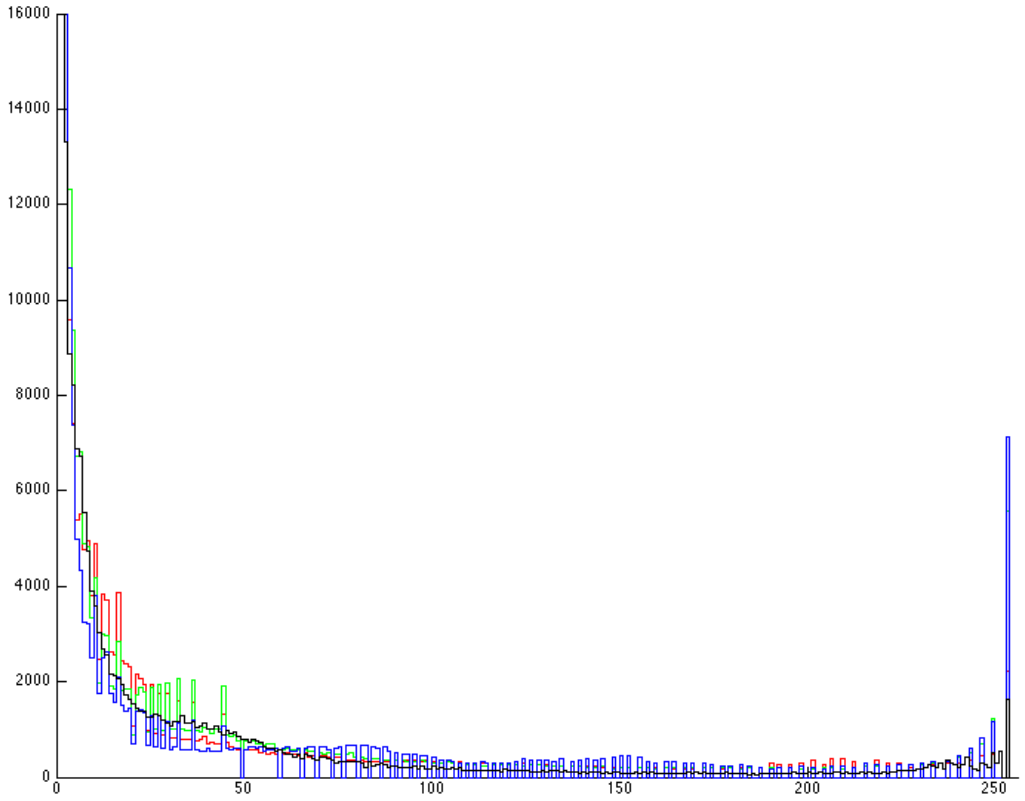

Figure 16. Histogram of Figure 15.

Experiment 3

b) - g) This experiment used piecewise linear lookup tables to selectively enhance certain parts of the image. In the first set of images the bottom third of brightness values were conrast enhanced, In the second, the middle third, and in the third, the top third. The lookup tables and PDFs are plotted along with each image.

Figure 17. Lower 1/3rd of pixel values contrast enhanced.

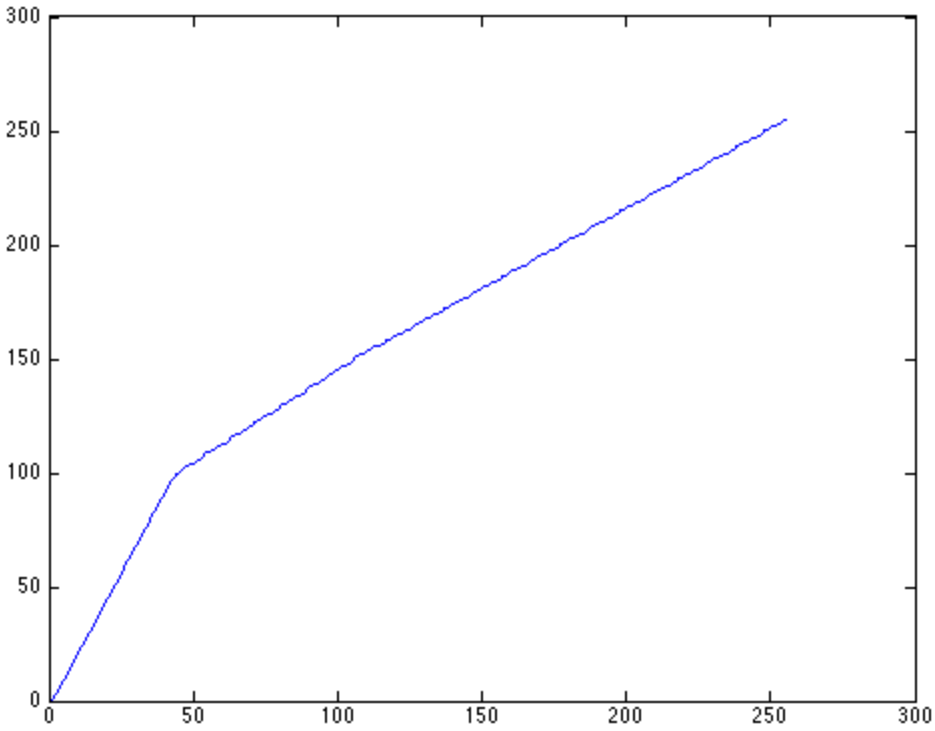

Figure 18. LUT used for Figure 17

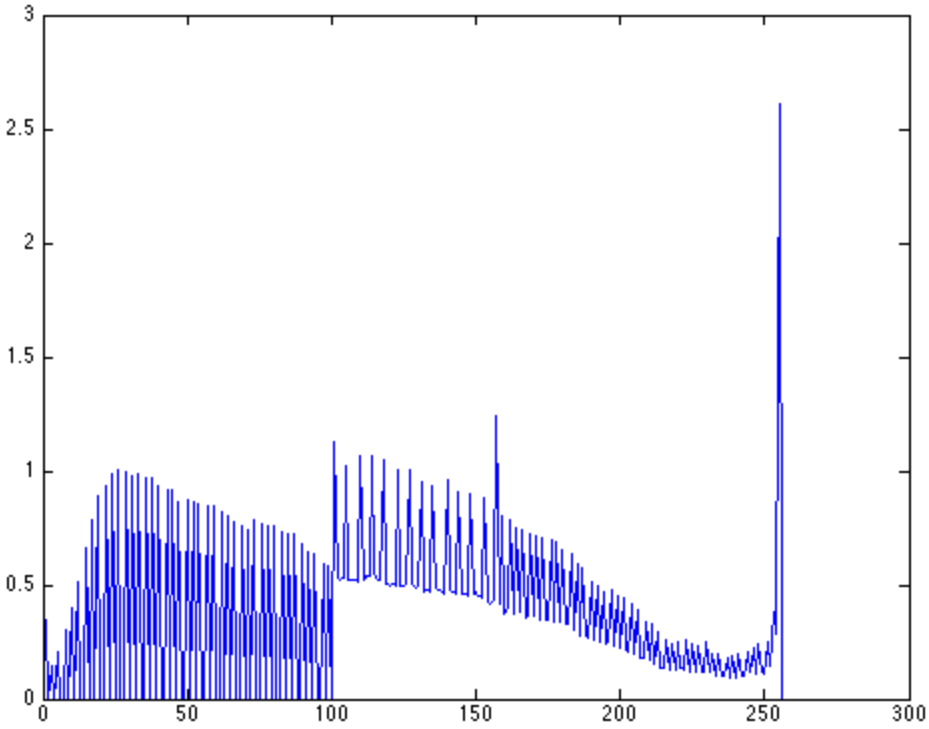

Figure 19. PDF of Figure 17

Figure 20. Middle 1/3rd of image contrast enhanced.

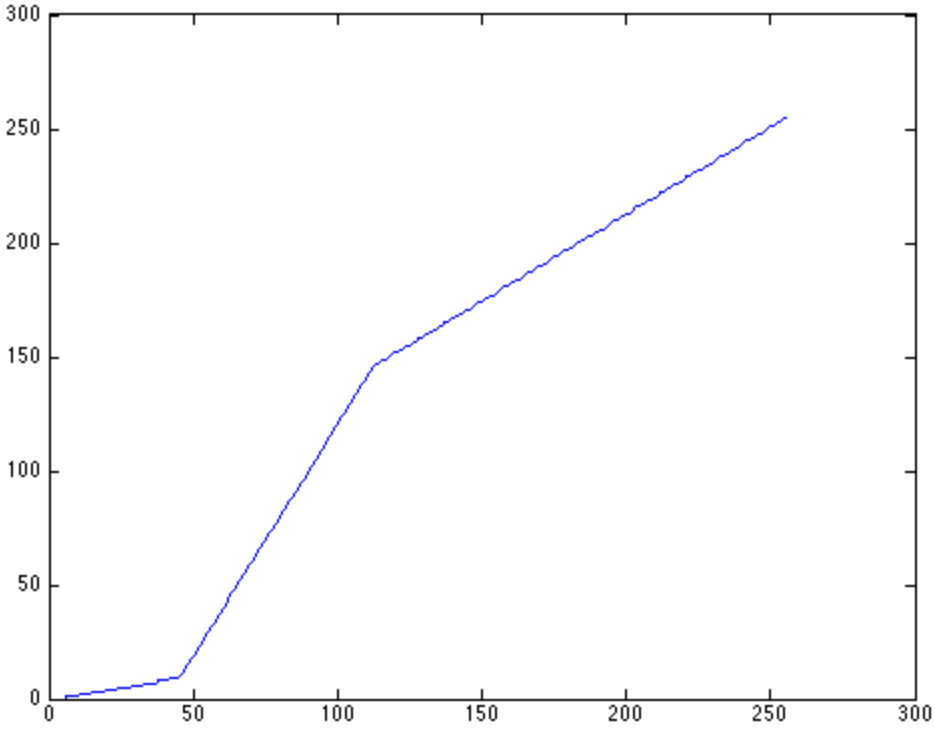

Figure 21. LUT used for Figure 20

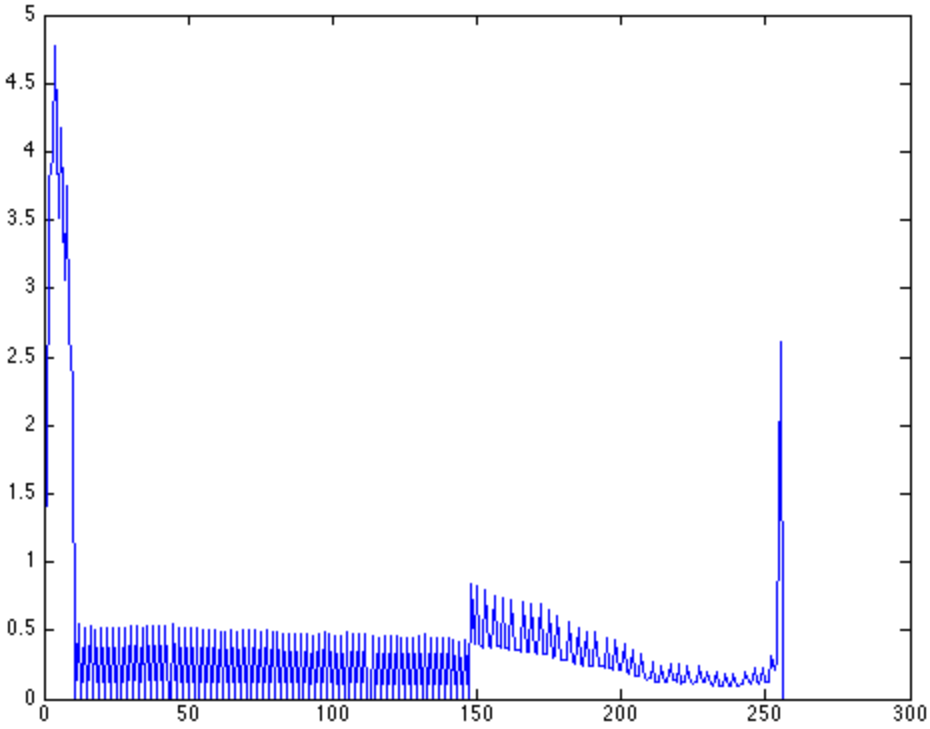

Figure 22. PDF of Figure 20

Figure 23. Upper 1/3rd of pixels contrast enhanced.

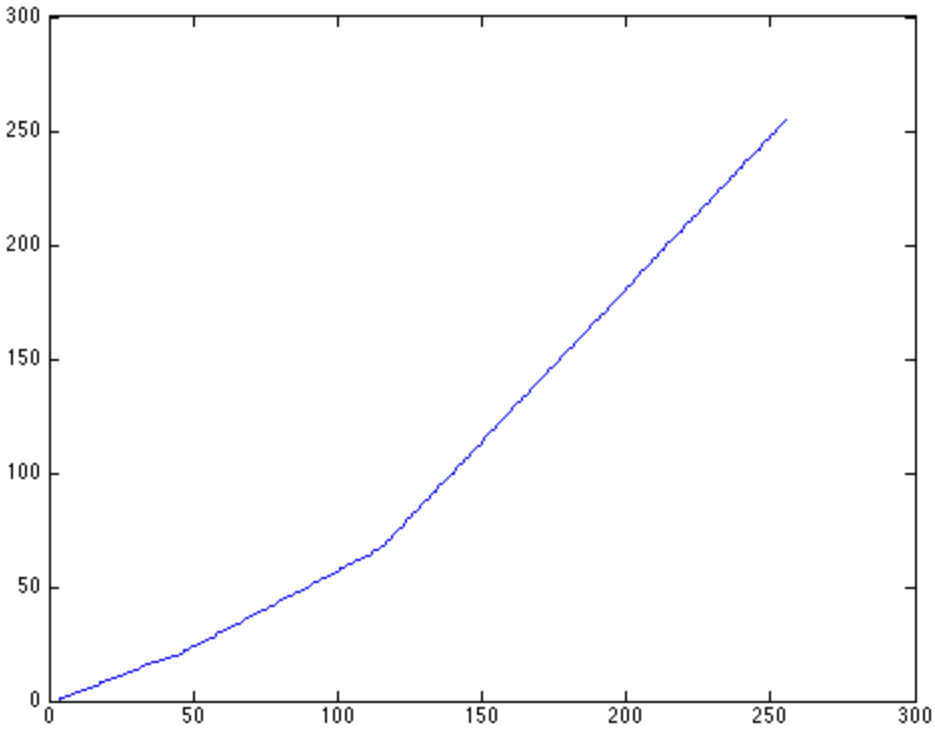

Figure 24. LUT used in Figure 23

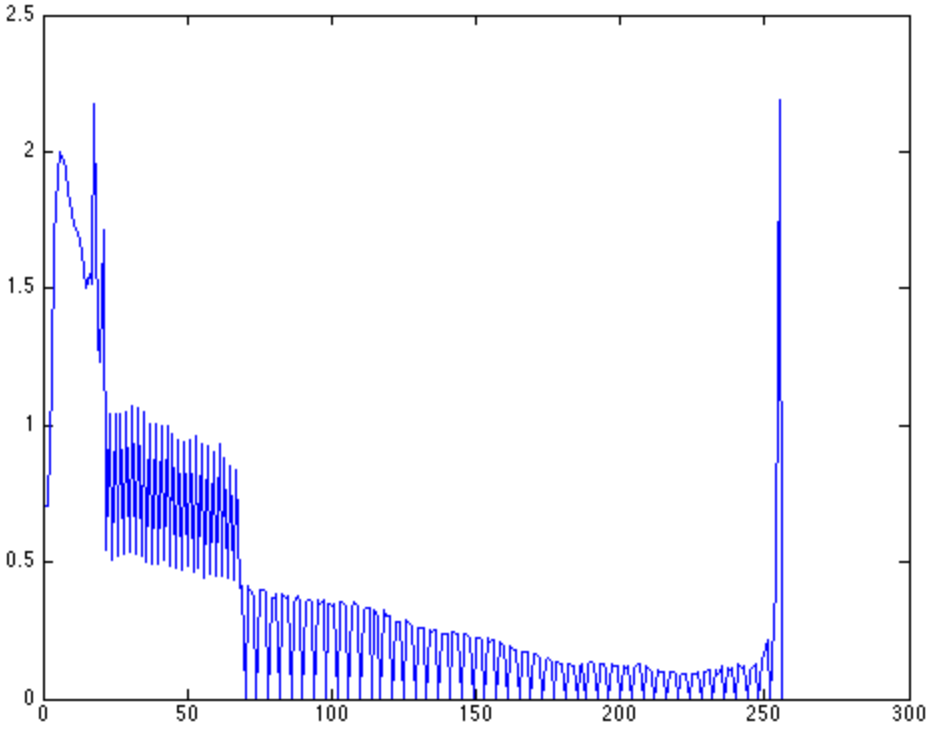

Figure 25. PDF of Figure 23

As you can see when the contrast in enhanced in the lower third, the shadows start to show more detail. When it is increased in the middle third, the middle values show more detail and when it is increased in the upper third, the bright areas show more detail. In the LUTs you can see the contrast enhanced region as a slope greater than 1 and a contrast decreased region as a slope less than 1. The script piecewise.m was used to generate and apply these LUTs. When you look at the PDF of each image, you can see the contrast enhancing effect shifts all the values in the given region.

Experiment 4

This experiment demonstrated methods for histogram equalization and histogram matching.

a) The program written to do histogram equalization is called histEQ.m

b) The output of the program on Figure 1 is shown below

Figure 26. Histograms equalized Figure 1

As you can see in the image, the contrast seems to be raised in all regions. The overall brightness has also be increased as compared to Figure 1. This is because all values were brought up to equalize the histogram.

d) Below if the PDF of Figure 26

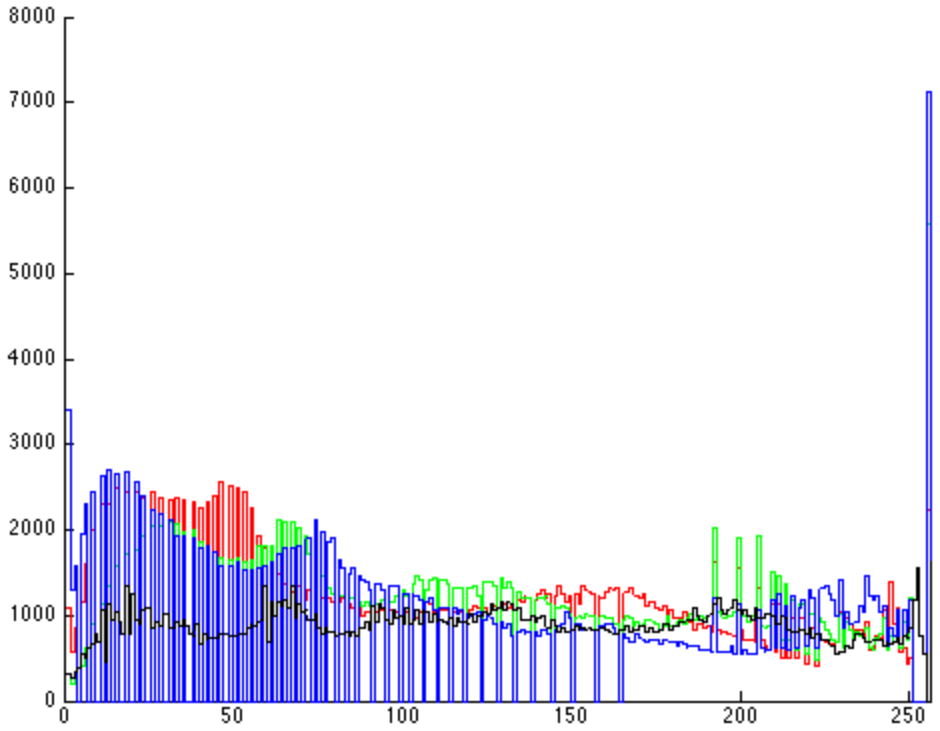

Figure 27. PDF of Figure 26

e) The program used to compute histogram matching did not work correctly. It always produced a grey image where every value was 87. You can see the attempted code in the file histMatch.m

Discussion of Results

Experiment one showed how to plot interesting features of an image, such as a histogram, probability distribution function, and a cumulative distribution function. These are useful tools to extract information from an image. Experiment two showed how to do simple point processing tasks such as brightness, contrast and gamma. These all can be accomplished using lookup tables, which map every value in the image to a corresponding value. A quick glance at the LUT can tell you much about the resulting image. Experiment three took LUT tables a step further by introducing piecewise linear LUTs. These can be used to selectively increase the contrast on certain regions of an image. And finally experiment four explored using histogram modification to change and image. It showed how to map one image to another, and how to equalize a histogram.

Conclusion

Each one of these experiments demonstrated different methods for point processing on images. Almost all of them utilized lookup tables to accomplish these transformations. These look up tables are a convient way to modify all the values of an image, as well as a good way to visualize what will happen to the image when the transformation is applied. The histogram was also explored as a means of examining the properties of an image, and possibly to be used for tamper detection. Using the histogram to manipulate the image was also explored, and this demonstrated the effects of equalizing the histogram as well as the effects of matching one image's histogram to another.