|  |
| --- |
| EECS 490 |
| Assignment 3 |
| Image Enhancement |
|  |
| **Edward Venator**  **Due: 9/29/2011** |
| **Handed In: 9/29/2011** |

|  |
| --- |
| This report examines the results of applying several histogram transformations, including contrast stretching, contrast to an image of the moon Phobos. |

# Technical Discussion

Many images can be improved with histogram transformations. Histogram transformations are point-by-point transformations, which means they are applied individually to each pixel of the image, without regard for the pixel's location or surroundings. In this project, I explored three kinds of histogram transformations: contrast stretching, histogram equalization, and histogram specification.

Contrast stretching is the process of taking a section of the histogram and stretching it to cover a larger range. This can help correct images that have a lot of pixels of similar value by "spreading" out these values over a larger part of the value spectrum.

Histogram equalization attempts to transform the image to create a flat histogram. This is accomplished by normalizing the histogram and integrating it to get a cumulative distribution function (CDF) of the image. The CDF is also a mapping from the current value of each pixel to a pixel value that will result in a flattened histogram. The result (which is then scaled from 0..1 to 0..255) will not be perfectly flat because of rounding errors, especially for input images with very uneven histograms.

Histogram specification takes the idea of histogram equalization one step further by specifying a specific histogram to transform to, rather than a flat histogram. The desired histogram is normalized and integrated to get a CDF. Using the image CDF, it is possible to create a map from a pixel value to desired pixel value using the following transforms:

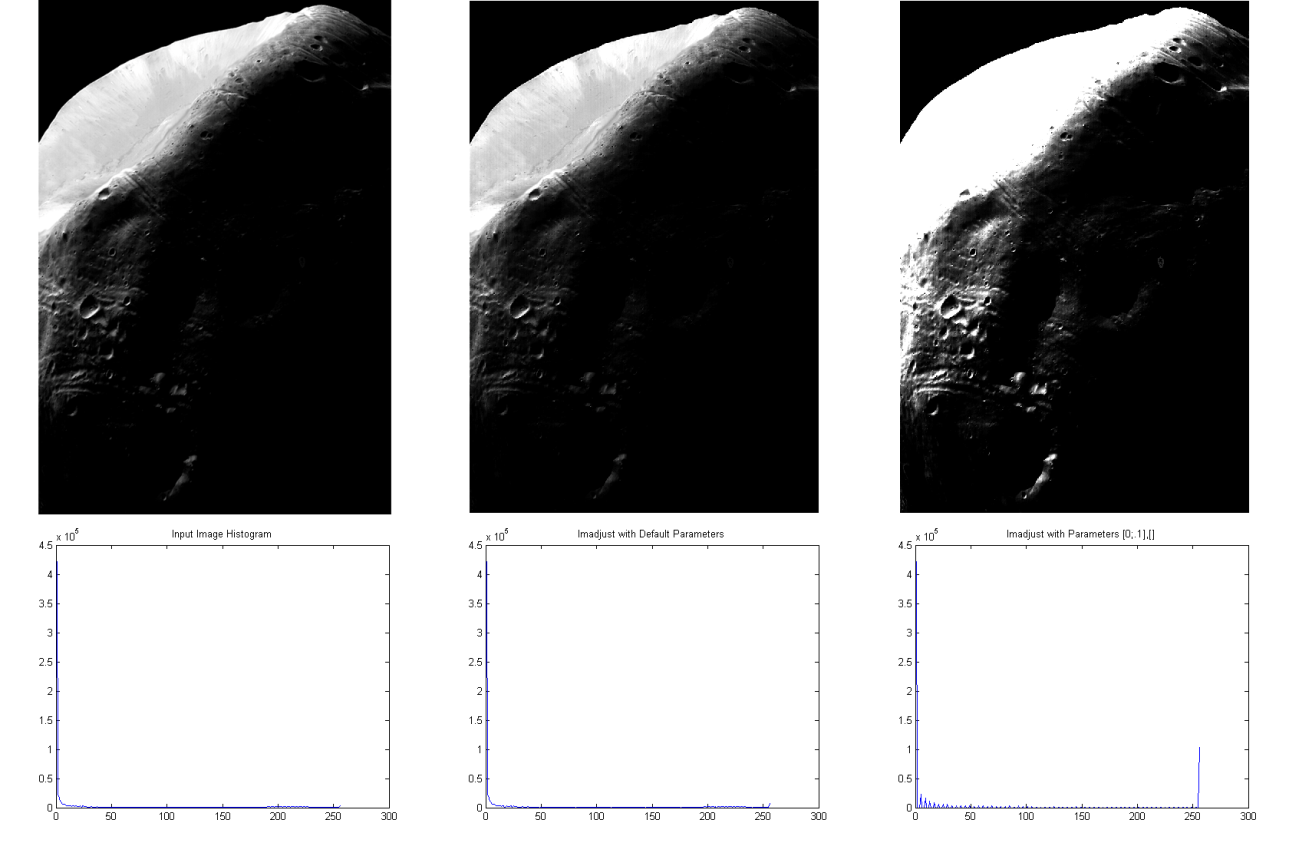
)

The easiest way to accomplish this is the calculate values of G for all possible input values (there are only 256 in an 8-bit image) and then create a lookup table. One limitation of the lookup table implementation is that it introduces rounding errors because the outputs of G must be rounded to integers to be used as lookup indices. My implementation uses a much slower search algorithm to lookup T(valuein) in G-1, but it should be slightly more accurate.

# Results

### Contrast Stretching

Contrast stretching had disappointing results. I implemented this using Matlab's imadjust() function, using both the default parameters and several other parameters.



*Figure 1: The original image and the results of contrast stretching with the default parameters and with my own parameters.*

Figure one shows the input image, the result of imadjust() with the default parameters, and imadjust() with one of the sets of parameters that I tested. Although the default parameters did not change the image much, I think that they resulted in a better image than any of the parameters I tested. My parameters performed more extreme contrast stretching, expanding the first tenth of the spectrum to fill the entire spectrum. An unfortunate side-effect is a lot of pixels are saturated at white, which destroyed a lot of detail in the brighter parts of the image.

### fig2.pngHistogram Equalization

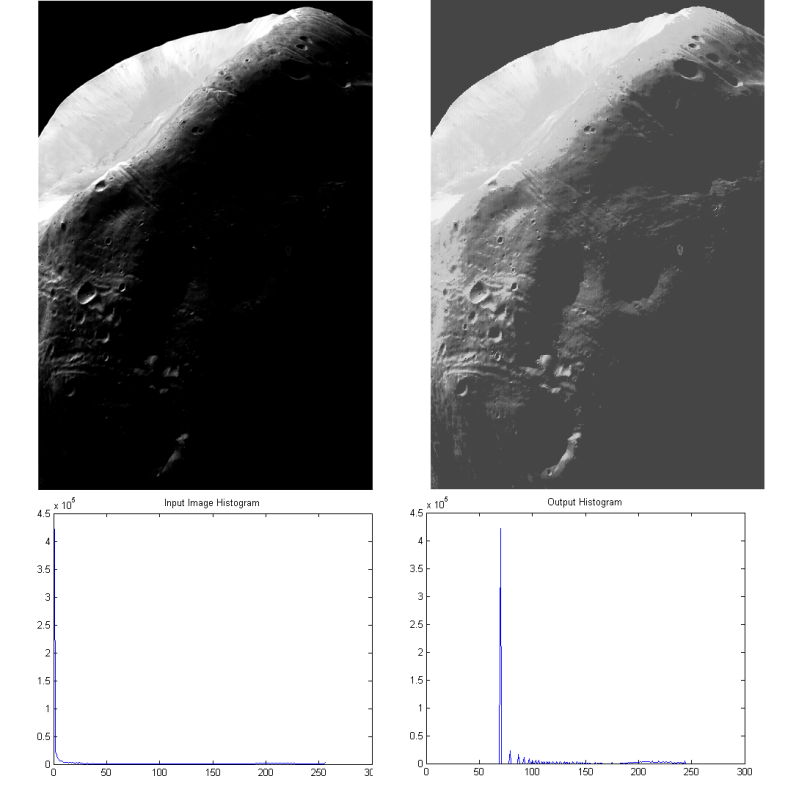
Histogram equalization resulted in better results than contrast stretching.

Figure two shows the input image and the result of histogram equalization, with their respective histograms. As you can see, the resulting image was not truly equalized. Because there were so many pixels with the same value (0 or black), there is still a massive spike, but it has been moved to the center of the value spectrum. The smaller "spikes" near it are caused by discretization (rounding) because so many of the values in the original image were so close together. The output image has "stretched" these sections, but because the input image is not continuous, there is no way to "fill in" between these pixel values, resulting in spikes. In my opinion, the biggest problem with this image is that it has shifted up in the value spectrum, making it very grey. Although this increases the visibility of many details in the dark parts, it also gives the image a "washed-out" appearance.

*Figure 2: The input image and the image after histogram equalization, with histograms shown.*

### Histogram Specification

Histogram specification was by far the most successful at improving the image. For my desired histogram, I chose to approximate the piecewise histogram that Woods and Gonzalez use on page 138 of the text.



*Figure 3: The result of histogram specification.*

As you can see in figure three, this method accentuates the detail without washing out the image like equalization did. My results are not as good as those achieved in the text, which is surprising when one considers that I specified almost exactly the same histogram. Their output histogram has a smaller spike than mine, and it is at a lower level. However, I am relatively pleased with this output. It shows the details of the dark parts of the moon without saturating the bright parts or washing out the entire image.

# Appendix

## Scripts

contrast\_stretching.m - Performs contrast stretching with several different parameters using MatLab's imadjust() function.

histogram\_equalization.m - Performs histogram equalization using Matlab's histeq() function.

histogram\_specification.m - Performs histogram specification using code written from scratch.

## Images

phobos\_X.tif - The input image used for all algorithms.

Output directory – All of the transformed images and histograms. Output images are in subdirectories based on which script generated them.