EECS: 332 Intro to Computer Vision, MP #3

Histogram Equalization, Due on 10/12/2017

Karan Shah

Introduction

Histogram Equalization is an image processing method which primarily uses the histogram of an image. It is used in order to increase the contrast of the image. In certain grayscale images, all the features are not clearly visible due to the excess of certain grayscale pixels. Therefore, in order to make the image have a better contrast, the pixels between 0 to 255 are distributed in a better manner. A cumulative sum function is useful in order to perform this operation.

Algorithm for Histogram Equalization

function [image_output] = Histogram_Equalization(image_input);

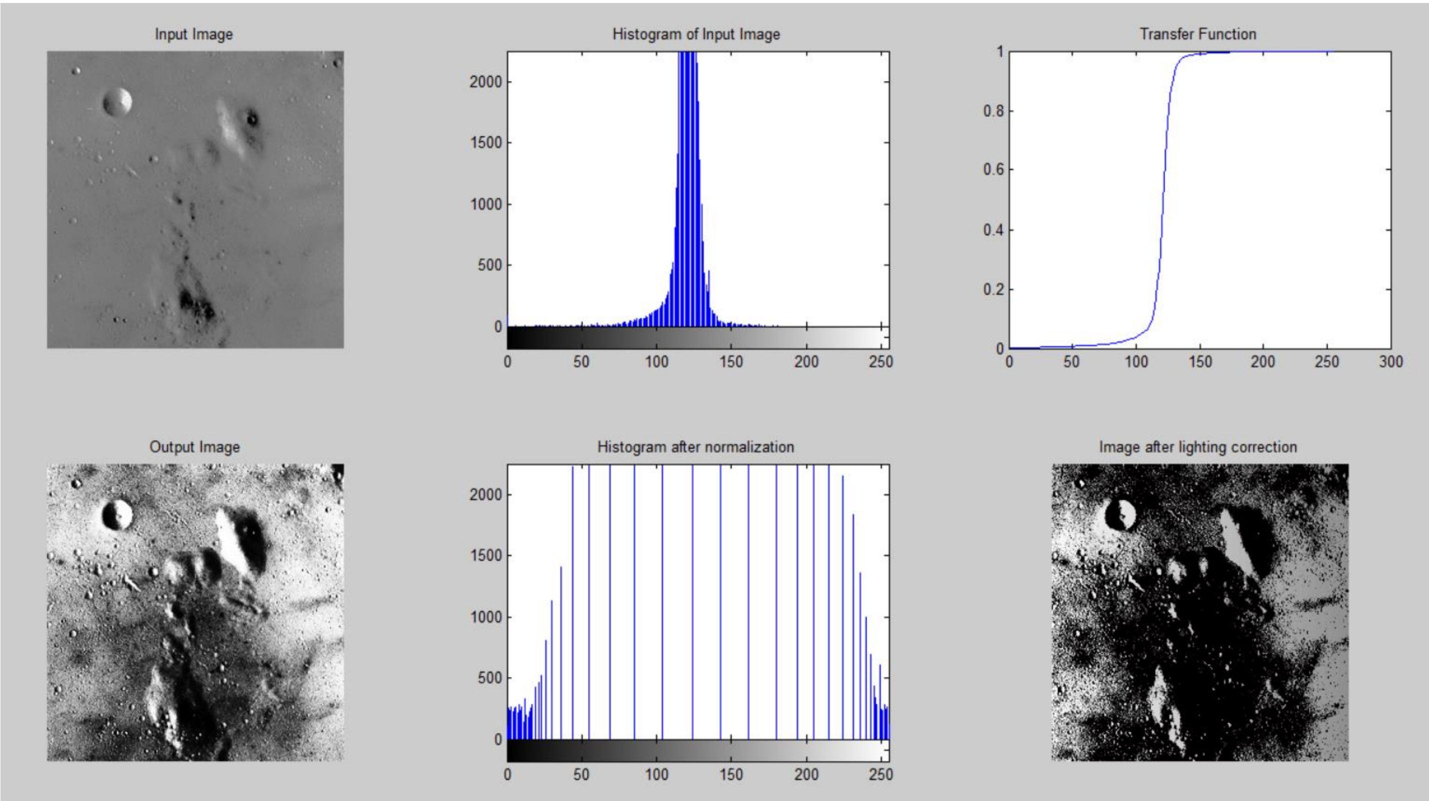
1. Create a Histogram for the Grayscale Input Image, by which, we have a table which holds values from 0 – 255 (Grayscale Levels), along with the no of pixels for each grayscale value. For ex. If an input image has 20 pixels of grayscale 0, 10 pixels of grayscale 1 and so on.
2. Next, we calculate a table which has the Probability Density (PDF) values. This table stores the information about each grayscale value and its relative no of pixels. For ex. If there are 4000 pixels, and there are 200 pixels of grayscale 0, then its relative value is 200/4000 which is 0.05. This is done for each grayscale value from 0-255.
3. After this, we calculate the Cumulative Density (CDF) values. This table stores the cumulative values of the previous table. For ex, in an input image, the Cumulative density table for grayscale 0 has just the probability density value for grayscale 0, but for grayscale 1, has both the values of 0 and 1. And this continues till grayscale value 255
4. Then, in another table, we multiply this cumulative density values with (256-1), as there are 255 values which are not zero.
5. After this, the values are rounded off to perform Quantization
6. This gives us the new histogram after equalization
7. After this, we iterate over the input image and check for the old Intensity levels and then replace it with the new intensity values by looping over two arrays
8. And then, the output image is created

Result Analysis

The results can be analyzed as follows:

1. Fig 1. Shows the Input Image, which is not very clear and not distinct
2. Fig 2. shows the Histogram of the Input Image
3. Fig 3 shows the Transfer Function
4. Fig 4. Which shows the Output Image (After Histogram Equalization) is much more distinct
5. Fig 5 shows the Histogram after Histogram Equalization
6. Fig 6 shows the Image after Lighting Correction

As you can see in Fig 1, the input image isn't very clear. But the output image in Fig 4 is very much clear and distinct. As you can see from the Histogram of the Input image in Fig 2, most of the pixels had an intensity between 100 and 150. Therefore, the image did not have a sharp contrast and all the pixels looked the same. However, in the histogram after equalization, you can see that the pixels are more evenly spread out from 25 to 225, therefore there is a much sharper image than the input image.



Algorithm for Lighting Correction:

1. Lighting correction works on least square regression. A group of data points, are represented in the form of a straight line is such a way that the least square distance from each of the data points to the line is minimum.
2. Therefore, the data set can be represented in the form of an equation a*X + b*Y + c = Z.
3. Therefore, for the given input image, we need to calculate co-efficient (a, b, c) to fit the line and data points. The x-axis co-ordinates of the input image are in X vector. The y-axis co-ordinates of the input image are in Y vector and pixel intensities in vector Z.
4. The inverse is found by treating the (ax+by+c = z) as a system, so the x and y co-ordinates are clubbed into the same matrix (X).
5. The vector is solved which has the co-efficients of (a, b, c) by employing the formula inv(X'*X) *X'*Y, where X is a matrix which is of the form [1 x (1) y (1); 1 x (2) y (2); ……] and Y is a matrix with the pixel intensities of the entire input image.
6. By using the Matrix method, a vector with the values of (a, b, c) is obtained.
7. Each pixel of the input image is obtained and then (ax+by+c) is subtracted from each pixel.
8. Thus, new image with lighting correction is obtained using Linear Method. The image is shown in Fig 6.