A5 – ENHANCEMENT BY HISTOGRAM MANIPULATION

Introduction

Digital cameras are designed to mimic properties of the human eye but there are some limitations in one that is not found in the other. For example, the sensor of a camera can record information under low light levels at which the human eye can not.

Consider for example the image below. Because of the bright background, the foreground turned out dark.

Figure 1. Picture of my research assistants taken aboard MY Navorca. The bright, sunny background caused the foreground to appear dark.

It may seem that information is lost in the dark parts of the image. The lady on the left is wearing a black shirt so one might think there is no reflected light from her shirt especially so that it is in the shadows (Figure 2). But as you can see in Figure 3 that by manipulation of the value output curve (done in GIMP), there is in fact information.

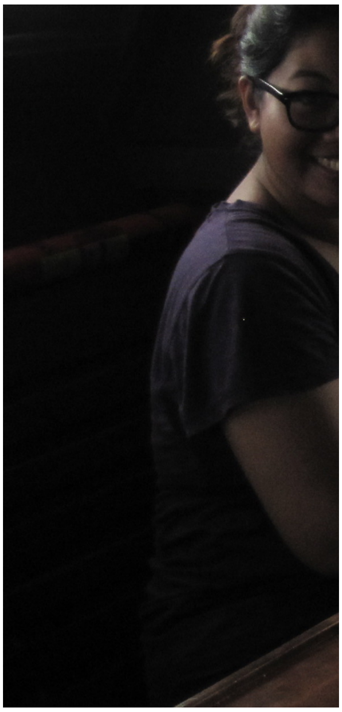

Figure 2. Dark portion cropped from left side of Figure 1.

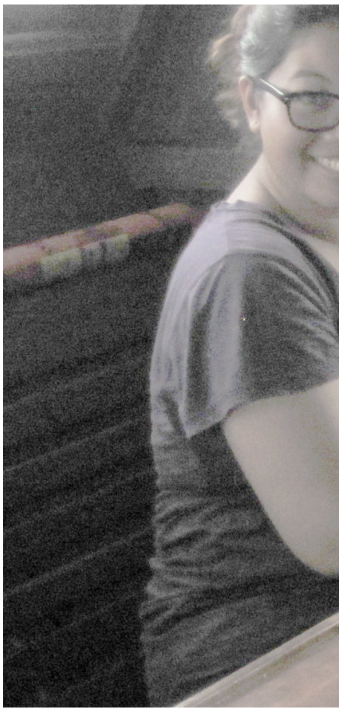

Figure 3. Enhanced image after value I/O curve manipulation using GIMP.

So do not trust your eyes. Clearly, the sensor of the camera picked up the information in the shadows except that our eyes are not sensitive enough to see them.

The histogram of a grayscale image is equal to the graylevel probability distribution function (PDF) if normalized by the total number of pixels. An image has poor contrast if its histogram is either skewed either towards the brighter or the darker graylevels. A good contrast image will have an evenly distributed histogram.

It is possible to improve the quality of an image by manipulating the histogram distribution. In this activity we will learn different techniques to alter the histogram of an image for a human observer to see information even in seemingly dark images. So don't delete those dark images yet.

Contrast Stretching

Suppose I_{min} and I_{max} are the minimum and maximum graylevels, respectively, of a poor contrast image. We can remap the pixel values $I(x,y)$ such that the minimum will have a value of 0 and the maximum can have a value of 1 by using the equation $I'(x,y)=\frac{I(x,y)-I_{min}}{I_{max}-I_{min}}$. You can then multiply the contrast-stretched value I'(x,y) by 255 if the image is 8-bit.

Histogram Manipulation by Backprojection

As such, the grayscale PDF of an image can be modified in the same way one can modify the PDF of random numbers. Histogram manipulation is one way of improving the quality of an image, enhancing certain image features, or mimicking the response of different imaging systems, such as the human eye.

Given the cumulative distribution function (CDF) of a desired PDF, the grayscale PDF of an image can be modified by ***backprojecting*** the grayscale values using the CDF.Suppose the graylevels *r* of an image has a probability distribution function (PDF) given by $p_1(r)$ and a cumulative distribution function $T(r)=\int_0^r p_1(g)dg$ where *g* is a dummy variable.

We want to map the *r*'s to a different set of graylevel *z*'s such that the new image will have a CDF given by $G(z)=\int_0^z p_2(t)dt$ where $p_2(z)$ is the PDF of the transformed image and t is a dummy variable.

If we let G(z) = T(r) then to get *z* we perform

$$z=G^{-1}[T(r)] \tag{1}$$

To illustrate, let Figure 4 be the grayscale PDF of an image and Figure 5 its CDF.

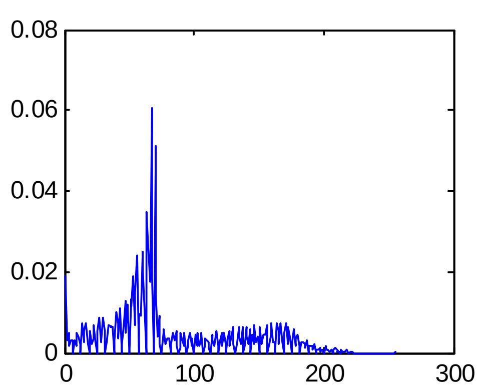

Figure 4: PDF (normalized histogram) of a grayscale image.

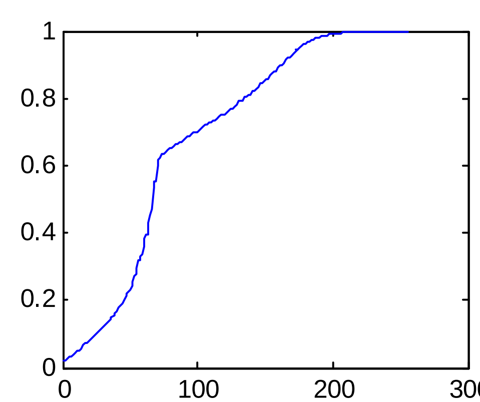

Figure 5: CDF of Figure 1 PDF.

Now suppose we want to remap the grayscales of the image such that the resulting CDF of the transformed image will look like that in Figure 6 below.

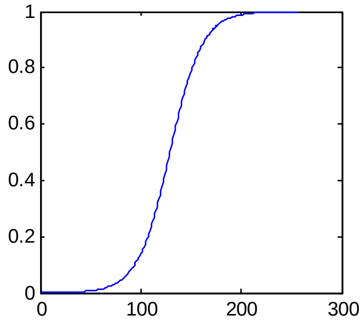

Figure 6: Desired CDF.

The steps are then:

1. Per pixel in the image having grayscale *r*, find its CDF value *T(r)*.
2. Find the value of *T(r)* in the y-axis of the CDF *G(z)*. Find the *z* value of this *G(z)*. See Figure 7.

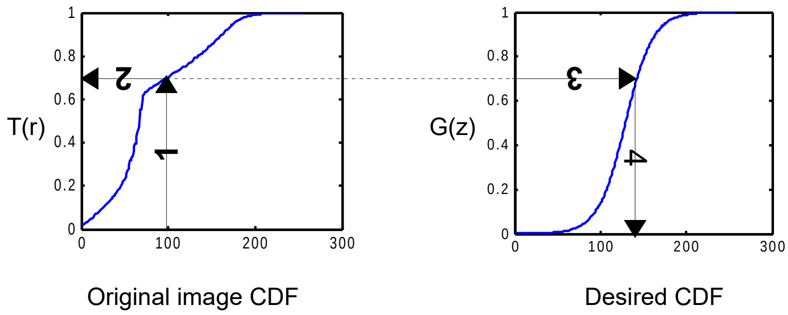

Figure 7: Steps in altering the grayscale distribution.(1) From pixel grayscale, find CDF value (2). Trace this value in the desired CDF (3). Replace pixel value by grayscale value having this CDF value in desired CDF (4).

Notes in Using Scilab and Matlab

Scilab and Matlab are optimized for matrix algebra. Each variable, even that representing a single number, is considered an array. Using **`for`** loops in Scilab or Matlab slows down calculations. To obtain faster computations, replace **`for`** loops with matrix operations.

Procedure

1. From your collection of images, pick a dark looking image and open it in Scilab or Matlab using **`rgb2gray`**. You may need to convert the variable to a double or float to avoid errors in mathematical operation handling.
2. Find the global minimum and maximum grayscale of your image and perform contrast stretching. Note that this technique will not work if your image has Imin = 0 and Imax = 255 simultaneously. Comment on the quality of the new image.
3. Determine the grayscale histogram of your image (use **`imhist or hist`**) . Find the CDF from its PDF and comment on its shape. Note, this will require knowing the histogram values and performing a cumulative sum. Check out **`cumsum`**.
4. The CDF of a uniform distribution is a straight increasing line. Create a CDF whose x-axis is the range of grayscale values of the original untreated image [0-1] and y-axis is equal to the value of x (or 0-255 if the pixel values are already in bytes, a higher range if the number of bits is higher).
5. Pixel-per-pixel, backproject the image pixel values by finding its corresponding y-value in the desired CDF. That is, replace the dark pixel values with the x-values from the desired CDF.
6. Display the modified image. This image is known as a histogram equalized image. Plot the histogram and get the CDF from the histogram values. Comment.
7. The human eye has a nonlinear response. Create a CDF that has a nonlinear shape (e.g. sigmoid function) and manipulate the histogram of your image.

