Assignment 2: Histogram Modification and Color Processing

Igli Duro
Department of Computer Science & Engineering
University of South Florida
Tampa, FL

I. Introduction

An image histogram is a representation of the brightness of pixels in an image. In this representation the number of pixels for each intensity value from 0-255 is plotted. And by doing so, we can observe the frequency and range of the intensity values that make up an image. For grey-scale images, it can be represented by a singular histogram. For color images, three histograms are needed to represent the red, green, and blue intensity levels in the composition. An image can be said to have low contrast if there's a small range of intensity values where the majority of pixels fall under. For this assignment we implemented processing techniques known as 'histogram stretching' which aims to correct and improve contrast in an image. For this report: Section 2 gives a description of each algorithm along with the the inputs they take, Section 3 goes into detail on the implementation of the algorithms, Section 4 will discuss our results, and Section 5 will contain our conclusions.

II. Description of Algorithms

In this section, in-depth explanations of the implemented algorithms are give and each algorithm will have its own subsection. The following algorithms in this section are: Histogram Stretching, Alternate Histogram Stretching, Histogram Thresholding, Histogram Color Stretching, RGB to HSI conversion, and an HSI Histogram Stretching algorithm.

A. Histogram Stretching

Histogram stretching is a technique used to improve contrast in an image. At a high level the idea is that we apply histogram stretching in order to spread out the intensity values more evenly across the range [0-255].

$$P' = (P - C)\left(\frac{B - A}{D - C}\right) + A$$

Fig. 1. Histogram Stretching Formula

Histogram stretching uses the formula in Figure 1. Where P' is our new intensity, P is the current intensity value, C & D are the min and max intensity of the image, and A & B are the desired intensities to stretch the image to. A & B are generally 0 and 255 respectively. But for our algorithm this is user specified. The algorithm will require two passes through every pixel in the image, the first pass to obtain a count of each intensity along with the min and max. The second pass then applies the stretching formula to each pixel. It should be noted that all proceeding algorithms are based off this technique with some modifications which will be mentioned.

B. Alternate Histogram Stretching

Histogram stretching on its own is a powerful tool capable of increasing image contrast, however it contains some limitations. If an image is low contrast but has a min and max intensity of 0 & 255, histogram stretching fails to improve it. In scenarios like this, an alternate stretching formula can be used. The formula itself is largely similar however we take the 5th and 95th percentile to use for our C and D. That is, we do C * 1.05 and D * 0.95.

C. Histogram Thresholding

Histogram thresholding is a technique that is used to segment an image into two subsets and perform stretching upon. The inputs for this algorithm take are: T, a threshold value, an input charter B or F that is used to specify 'foreground' or 'background' and input values A and B, which are our desired intensities. With this technique, we first perform thresholding on an image to create a group of pixels. If a pixel intensity is above the threshold we set its value to 0, otherwise we set it to 255. Along with this we track the min and max for both foreground and background group. Afterwards we perform histogram stretching on the foreground or background group depending on the input.

D. Histogram Color Stretching

Color stretching is a technique that uses histogram stretching. As such it is largely similar, with the introduction of multiple color channels. The color stretching algorithm operates by taking in a specified input: R, G or B. Based off that input the algorithm performs histogram stretching only on the specified channel. The other two channels remain unmodified. This method produces less intense results comparative to coloring brightening which multiplies the color channel by a constant value.

III. Description of Implementation

For the implementation of the histogram stretching algorithms we built off our preexisting image processing toolkit. Which included the image and utility class. For our utility class we added functions for histogram stretching, alternative histogram stretching, threshing histogram stretching, and color histogram stretching. Each function is largely similar to the base histogram stretching algorithm with some slight variations. For the histogram stretching algorithm we accept four inputs, the source image, target image, A, and B. Where A and B are the min and max desired intensities to stretch to. The algorithm makes two passes, looping over every pixel in the source image. The first pass is to find the min and max intensity that will be used in our formula. Along with the frequency of each intensity. The second pass applies the formula to each pixel and saves it to the target image. The alternative histogram stretching functions the same, but uses a slightly modified min and max value for the image. Using min*1.05 and max*0.95 for the calculation.

The thresholding histogram algorithm variation is slightly different and accepts two additional inputs. A user inputted threshold and the character F or B to signify foreground or background. The function groups the pixels into two categories in relation to the threshold then tracks the min and max for the foreground or background specifically. It then only applies stretching to that region.

The color histogram stretching algorithm accepts the characters R, G, or B, and a user defined A and B. It then finds the min and max of that specified color channel and applies stretching to it. The other remaining channels remain the same and are copied over.

Fig. 2. mountain.jpg

Fig. 3. mountains.pgm mountains1.pgm 1 0 0 408 612 histostretch 0 255

IV. Description of and Analysis of Results

A. Description of Results

This section will go over the results of the algorithms implemented. We'll first be looking at the histogram stretching called and alternative stretching algorithm. Followed by the thresholding histogram stretch and lastly the color histogram stretching algorithm.

Figures[2-3] show the results of using histogram stretching on the 'mountains.jpg' file. Comparing the before and after we can see increased contrast throughout the image. Where the base image was full of medium gray tones. The output uses a wider range of intensities. Darker grays became more black and lighter grays became more white.

Next in Figures[4-7] we see the results of applying multiple streching algorithms onto castle.jpg. Figures 5 and 6 are a comparison between histostrerch and althistostrech. From it we can see some slight differences and Figure 6 is slightly more brighter. This is expected as althistostrech uses slightly higher and lower bounds for the min and max so the resulting image becomes a bit more gray. Figure 7 shows histrogram stretching across 3 different regions of interest. Segmenting areas of an image can result can produce different results as opposed to applying the filter to the entire image. As each segment may have a different min and max intensity which will affect the resulting stretch.

In Figures[8-10] we see the result of the thresholding and histogram stretching applied to the 'slope.jpg' file. In Figure 9 we can see the foreground pixels with intensities greater than 250 get selected and stretched. And likewise for figure 10 we can see pixels less than 50 get selected for the background.

For Figure[11] we apply color histogram stretching onto the globe.jpg file. For this 3 ROIs were selected and in each one a different color channel is stretched. This produced interesting results as the globe itself was colorful but was surrounded by a gray background. Resulting in different color tones for the greys being stretched vs colored portions being stretched. When red is stretched the gray portions become cyan. When blue is stretched the gray portion becomes yellow. And when green is stretched the gray portions become magenta. This unintentionally became something of an RGB to CYMK conversion. As when intensities are being stretched, in a way its being subtracted. As they are being spread out. Cyan is produced when red is subtracted. And likewise for yellow and magenta when blue and green are subtracted respectively.

Fig. 4. castle.jpg

Fig. 5. castle.pgm castle1.pgm 1 0 0 564 533 histostretch 20 235

Fig. 6. castle.pgm castle2.pgm 1 0 0 564 533 althistostretch 20 235

Lastly in Figures[12-15] I took a picture of an RGB lamp I have which when the LEDs were set to red. I wanted to attempt producing white in the image by doing multiple passes of stretching out the red, blue and green channels. When stretching out the blue channel magenta tones are produced. And when green is stretched on top of that we begin to get some yellow tones where red was intense and cyan tones where magenta was intense. I didn't manage to get the desired result but the overall image was indeed becoming brighter.

V. Conclusion

For this assignment, we implemented a set of histogram stretching algorithms. From these newly implemented algorithms we gained the functionality to improve contrast in both gray scale and color images. Adding onto our toolkit are a set of different stretching algorithms that are useful in a variety of situations. And for our results overall we were able to obtain desired results.

Fig. 7. castle.pgm castle_hs3roi.pgm 3 0 0 300 200 histostretch 0 200 0 200 300 333 histostretch 0 150 300 0 264 533 histostretch 20 255

Fig. 8. slope.jpg

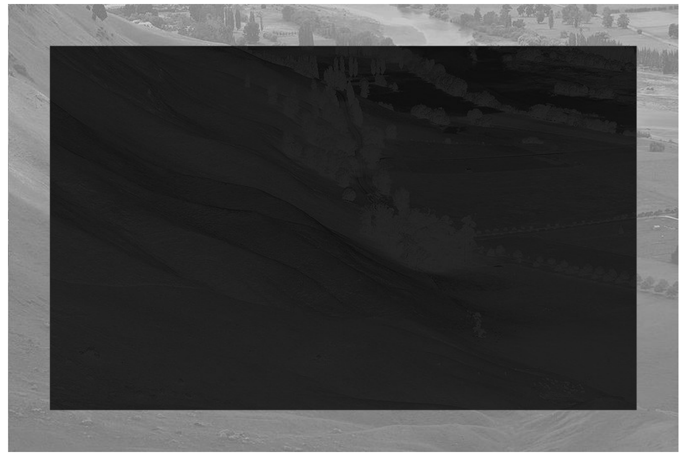

Fig. 9. slope.pgm slope_HSTF.pgm 1 50 50 434 700 histothres 250 F 0 75

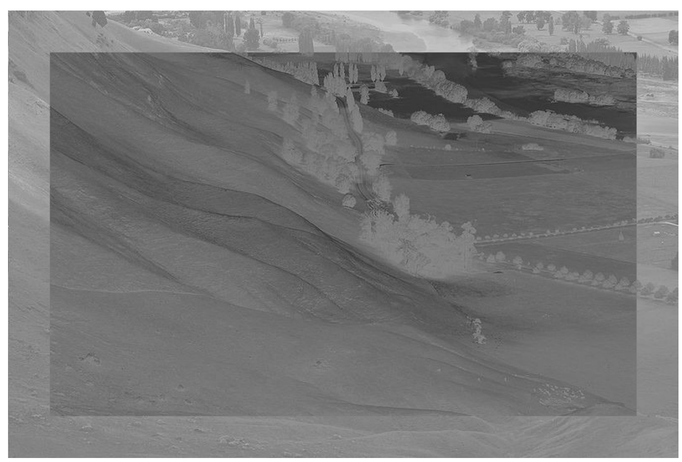

Fig. 10. slope.pgm slope_HSTB.pgm 1 50 50 434 700 histothres 50 B 0 255

Fig. 11. globe.ppm globe_3roi.ppm 3 50 50 400 100 colorstretch R 125 200 300 300 150 150 colorstretch G 50 200 100 200 300 300 colorstretch B 50 200

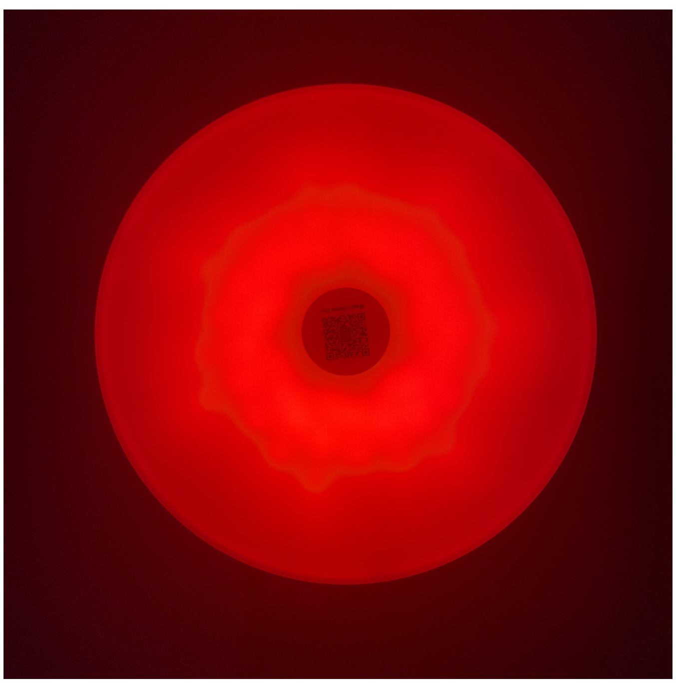

Fig. 12. redlight.jpg

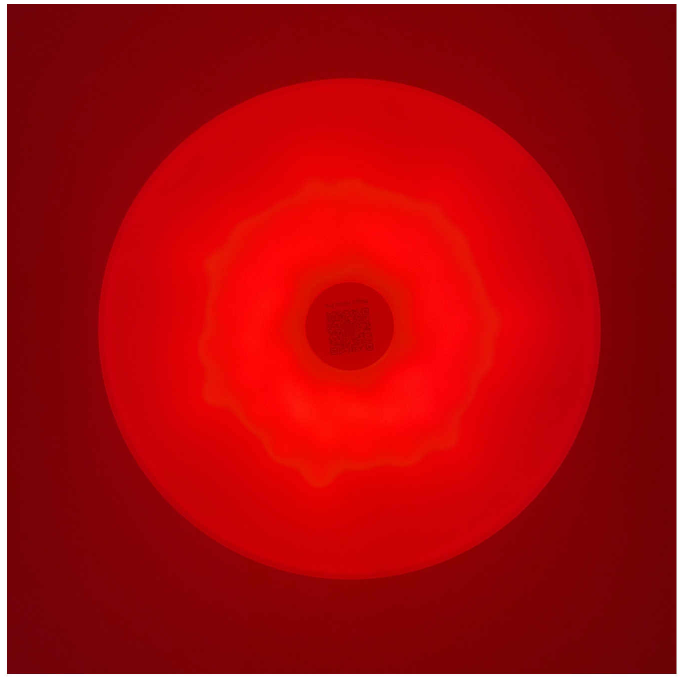

Fig. 13. redlight.ppm redlight_r.ppm 1 0 0 3024 3024 colorstretch R 100 255

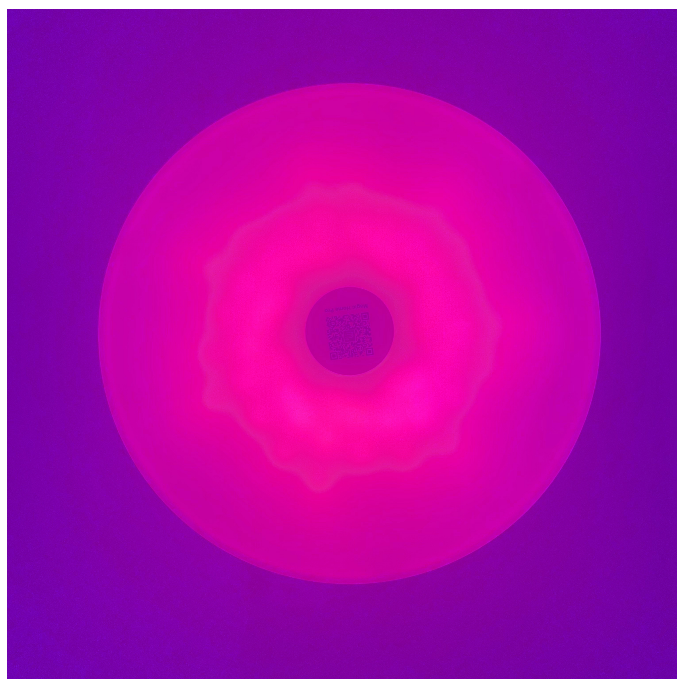

Fig. 14. redlight_r.ppm redlight_rb.ppm 1 0 0 3024 3024 colorstretch B 150 255

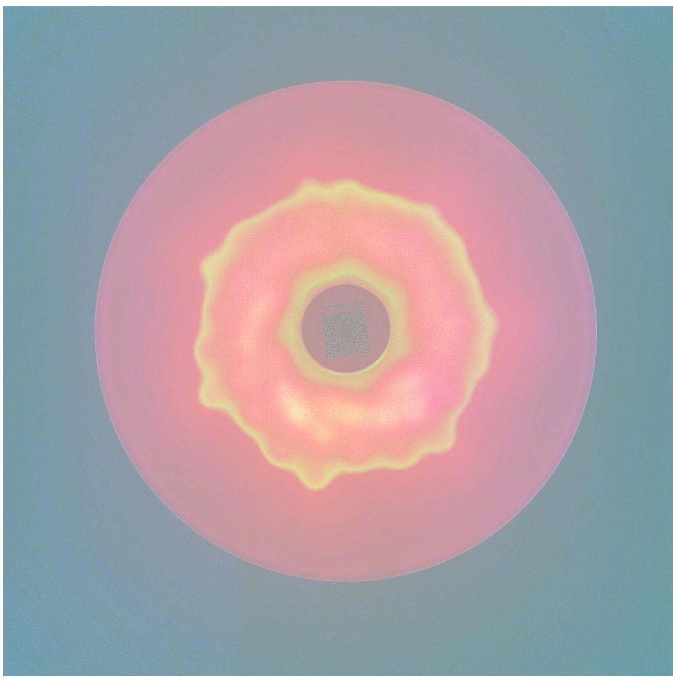

Fig. 15. redlight_rb.ppm redlight_rbg.ppm 1 0 0 3024 3024 colorstretch G 150 255