CAP5400 – Digital Image Processing

Assign 2:
Histogram Modification and Optimal Thresholding

Name: Zhenglong Wu
U#: U96342243
Date: Oct. 7th, 2019

1. Introduction

This assignment mainly focuses on the histogram stretching in gray-scale image and color image to implement contrast enhancement, as well as calculating the optimal threshold for image binarization. And all image processing operations must be performed in the specified non-overlapping region of interest (ROI). Image signals generally contain a lot of information, if there are important information concentrated on similar gray scales, it is difficult for the human eye to identify them. In this case, the histogram of image is represented as a mountain peak (unimodal or multiple peaks (bimodal or multimodal). Histogram stretching can make darker pixels darker and brighter pixels brighter in a certain area to achieve contrast enhancement. Thus, the human eye can more easily identify the details in the image. Unlike the adaptive threshold of local operation, optimal thresholding is an iterative method in global operation. Optimal thresholding is based on the assumption that the image has a bimodal histogram, so objects can be extracted from the background by a simple operation that compares the image values to the threshold. Therefore, for images with distinct bimodal peaks and deeper valleys, the optimal threshold (iterative method) can obtain satisfactory results faster. In the subsequent experiments, Section 2 describes the interpretation of some algorithms. Section 3 describes the operations and implementation details, such as parameters and all run options. In section 4 and 5, the results of experiment and conclusion are presented.

2. Description of algorithm

In this section, there are some explanations of the new algorithms encountered in the experiment. The description of ROI setting method will be omitted, which is to manually set the location and size of the ROI in the image. The conversion between HSI and RGB is based on the official formula. It is not the main purpose of this experiment and will not be detailed.

2.1. Histogram stretching

Histogram stretching algorithm needs two parameters a and b, which are usually the minimum intensity (a) and maximum intensity (b) of the image gray-scale level. The algorithm stretches the aggregated pixels of the original picture in the interval of (a, b) to (0, 255), where the brighter pixels in the interval will become brighter and the darker pixels will become darker.

There is another stretching algorithm -- bilinear histogram stretching. It needs four parameters a, b, c and d. Among them, a and b have the same meaning as the previous one, c and d represents stretching or compressing the interval of (a, b) in the original image to the interval of (c, d) in the new image.

2.2. Optimal thresholding

In this experiment, optimal threshold algorithm uses the median intensity as the initial threshold. Then this threshold divides the image into two parts: a

background which represents all pixels larger than the threshold and an object which represents all pixels smaller than the threshold. After we get two sets of pixels: the background and the object, we start to calculate the mean of the mean value of the two sets as a new threshold. If the difference between the new threshold and the previous threshold is less than a certain limit, such as 0.5, then the optimal threshold is the latest threshold. Otherwise the new threshold will be used to continue the iterative operation until the optimal threshold is found.

2.3. Combined optimal thresholding and histogram stretching

In this experiment, optimal threshold algorithm and histogram stretching algorithm is combined. Firstly, we use optimal threshold to segment background and object. Secondly, we analyze the histogram of each background and object, then separately stretch both of them into range (0, 255) and eventually merge. In the histogram stretching operation, a and b are the minimum intensity and maximum intensity of each subregion.

2.4. Color image HSI color space histogram stretching

For color image, HSI color space is easier to image processing than RGB. This is because HSI stores all the colors that the human eye can recognize in Hue, while the other two scalar Saturation and Intensity can control the shade and brightness of the color in white. Therefore, we can manipulate the S and I components separately without worrying about the H color component. However, the logic of RGB is to divide all the colors that can be recognized by the human eye into three primary colors, red, green and blue. And new colors are obtained by controlling the brightness of the three primary colors and different mixing ratios. When we manipulate the brightness of the three components of RGB separately, the blending ratio of the R, G and B will also change, causing a partial distortion of the final color.

In summary, the human visual system is more sensitive to changes in brightness than color changes. HSI is more suitable for image processing in line with human vision systems, and RGB is more suitable for storing and generating images.

In this experiment, the RGB image was changed to HSI representation at first, and after Histogram stretching and other processing, the HSI is re-converted to RGB for image displaying. The algorithm in histogram stretching part is the same as before, and the description will not be repeated here.

3. Description of implementation

The entire codes are developed in the MatLab. In parameter file, there are four common parameters for each image processing, which are “input filename”, “output filename”, “operation name” and “ROI location and size”. In addition, some operations needs more parameters, such as the user-defined gray-scale range in histogram stretching. The code reads the parameter file and run the specified functions to get the output image.

parameter.txt – Common section			
Input filename	Output filename	Operation name	ROI location & size
lena.pgm	lena_histStretch.pgm	histStretch	[28,28…;…;…456]
…	…	…	…

Each ROI is defined by 4 parameters: Rx, Ry, Sx, Sy. Rx and Ry is the location value in both x-axis and y-axis of the top left corner of the ROI, Sx and Sy is the size value in width and height of the ROI. The format of ROI parameters is [28,28,200,200; 28,284,200,200; 284,28,200,456]. It is a Nx4 matrix, which has N rows, and each row represents a ROI in the image. And four columns represent Rx, Ry, Sx, Sy separately.

When the code read the ROI, it will also run an overlapping test to ensure there are no two ROIs overlaps in the same image.

parameter.txt – Optional section			
Operation	Parameter1	Parameter2	…
histStretch	[60,180;33,230;28,220]	…	…
…	…	…	…

For example, parameter1 is one of the parameters required by operation “histStretch”, which represents the minimum intensity (a) and maximum intensity (b) of the gray-scale level in each ROI.

4. Description of results

4.1. Histogram stretching of each ROI

Figure 1.1

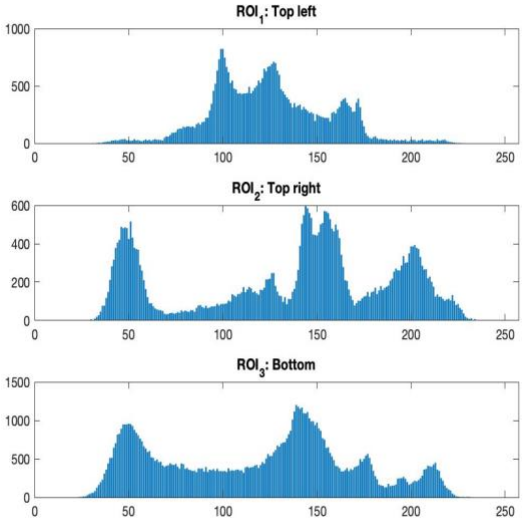

Figure 1.2

Figure 1.3

RO1_1 (a,b): (60, 180)

RO1_2 (a,b): (33, 230)

RO1_3 (a,b): (28, 220)

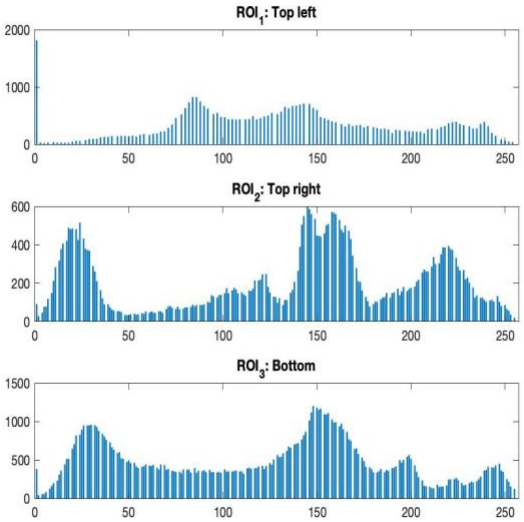

Figure 1.4

Histogram of each ROI

The figure 1.1 shows the three ROIs that we choose in the experiment, and the figure 1.2 is the corresponding histogram of each ROI. The figure 1.3 shows the images after the histogram stretching operation on all ROIs, which based on the formula:

$$I_{(x,y)} = \begin{cases} 0 & if\ I_{(x,y)} \le a \\ \min\left(255,\ \ (I_{(x,y)} - a) * \dfrac{255}{(b-a)}\right) & if\ a < I_{(x,y)} \le b \\ 255 & if\ I_{(x,y)} > b \end{cases}$$

The top left ROI_1 stretches all pixels with the gray-level between 60 and 180. Similarly, the top right ROI_2 stretches the gray-level between 33 and 230, the bottom ROI_3 stretches the gray-level between 28 and 220. The figure 1.4 shows the new histogram of each corresponding ROI.

4.2. General bilinear histogram stretching

Figure 2.1

RO1_1 (a,b,c,d): (100,150,50,205)
RO1_2 (a,b,c,d): (130,170,50,205)
RO1_3 (a,b,c,d): (100,150,50,205)

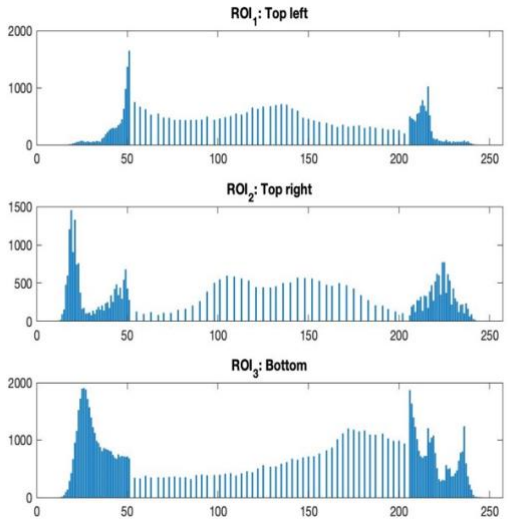

Figure 2.2
Histogram of each ROI

The figure 2.1 shows the images after the bilinear histogram stretching operation on all ROIs, which based on the formula below:

$$I_{(x,y)} = \begin{cases} \min\left(255, \quad I_{(x,y)} * \frac{c}{a}\right) & if\ I_{(x,y)} \leq a \\ \min\left(255, \quad \left(I_{(x,y)} - a\right) * \frac{(d-c)}{(b-a)} + c\right) & if\ a < I_{(x,y)} \leq b \\ \min\left(255, \quad \left(I_{(x,y)} - b\right) * \frac{(255-d)}{(255-b)} + d\right) & if\ I_{(x,y)} > b \end{cases}$$

The top left ROI_1 stretches pixels from the gray-level between range (100, 150) to range (50, 205), the remaining pixels are compressed to range (0, 50) and range (205, 255). Similarly, the top right ROI_2 does the bilinear histogram stretches in the gray-level from range (130, 170) to range (50, 205), the bottom ROI_3 does the bilinear histogram stretches in the gray-level from range (100, 150) to range (50, 205). The figure 2.2 shows the new bilinear stretched histogram of each corresponding ROI.

4.3. Optimal thresholding (Iterative thresholding)

Figure 3.1

Figure 3.2

RO1_1 (iterations, threshold): (4, 126)
RO1_2 (iterations, threshold): (1. 147)
RO1_3 (iterations, threshold): (1, 128)

The figure 3.1 is the original image, and the figure 3.2 is the binarization operation after using optimal threshold in each ROI. The optimal threshold is based on the formula below. I represents all pixels in the image. H_o and N_o are the object pixels set and the

number of pixels in object set. H_b and N_b are background pixels set and number of pixels in background set.

$$T_{initial} = Median\ of\ I$$

$$\begin{cases} H_o = I_{(x,y)}[\ I_{(x,y)} \leq T\ ] \\ H_b = I_{(x,y)}[\ I_{(x,y)} > T\ ] \end{cases}$$

$$newT = \left(\frac{\sum H_o}{N_o} + \frac{\sum H_b}{N_b}\right) * \frac{1}{2}$$

If the difference between "new T" and "initial T" is greater than 0.5, then assign "new T" to "initial T" and then recalculate the new "new T" until the difference between them is less than or equal to 0.5. The figure 3.2 In the optimal thresholding result, ROI_1 gets threshold 126 in 4 iterations, ROI_2 gets threshold 147 in 1 iteration, and ROI_3 gets threshold 128 in 1 iteration. Comparing these thresholds with Figure 1.2 histogram, these thresholds all tend to be in the middle of two double peaks, trying to distinguish between background and objects.

4.4. Combine the background and object histogram stretching

Figure 4.1

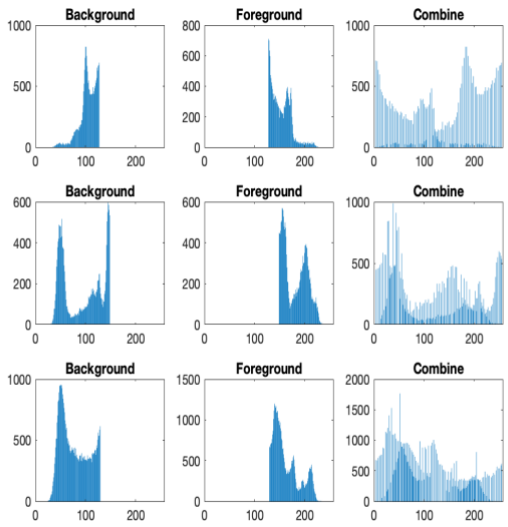

Figure 4.2

The figure 4.1 is the images that combine the stretching of both background and object (foreground). The formula to implement this processing is:

$$\begin{cases} T = optimal\ threshold \\ a = \min(I) \\ b = \max(I) \end{cases}$$

$$I_{(x,y)} = \begin{cases} 255 - (\mathrm{T} - I_{(x,y)}) * \left(\frac{255}{T-a}\right) & if\ I_{(x,y)} \le T \\ (I_{(x,y)} - \mathrm{T}) * \left(\frac{255}{b-T}\right) & if\ I_{(x,y)} > T \end{cases}$$

T is the optimal threshold of a ROI we calculate by iterative threshold algorithm. Then a and b are the minimum and maximum intensity of the image. Therefore, in the object subregion, a and T are the minimum and maximum intensity. Similarly, T and b are the minimum and maximum intensity in the background subregion.

4.5. Color image histogram stretching

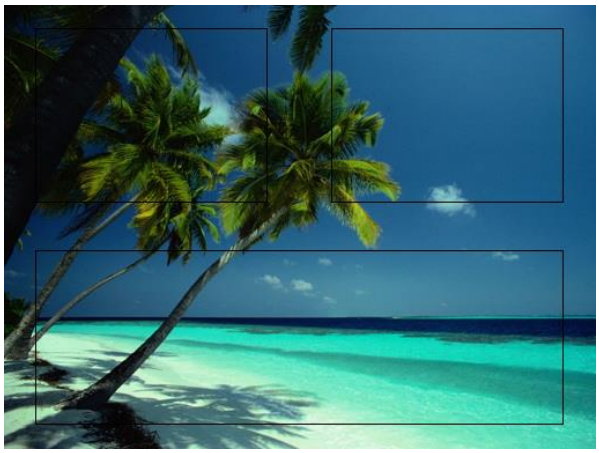

Figure 5.1 Original Image

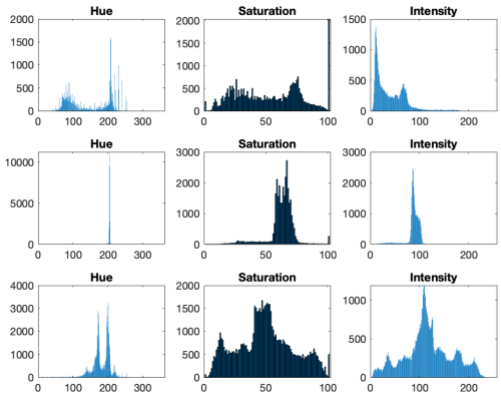

Figure 5.2 Hue, Saturation, Intensity histogram of figure 5.1

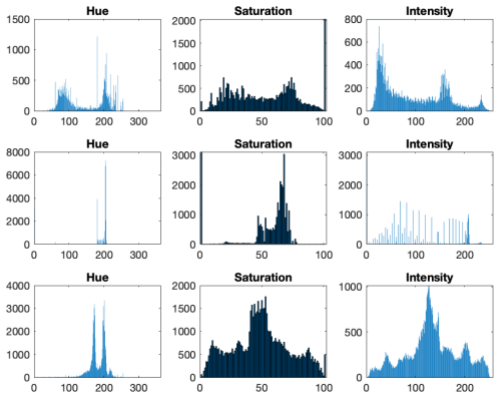

Figure 5.3 Intensity Stretching Image

Figure 5.4 Hue, Saturation, Intensity histogram of figure 5.3

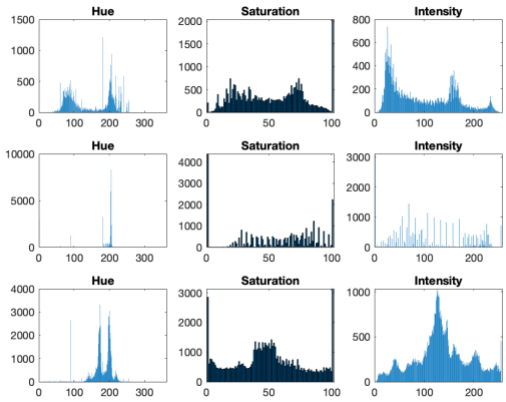

Figure 5.5 Saturation and Intensity Stretching Image

Figure 5.6 Hue, Saturation, Intensity histogram of figure 5.5

Figure 5.7 Hue, Saturation and Intensity Stretching Image

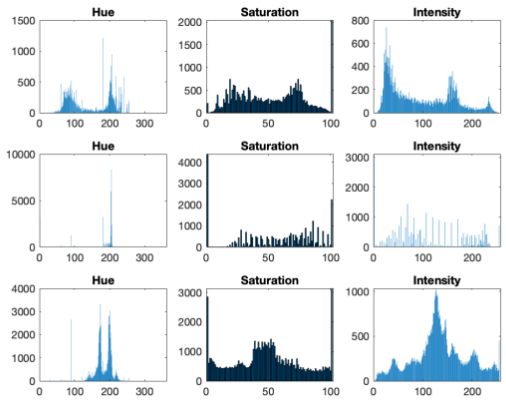

Figure 5.8 Hue, Saturation, Intensity histogram of figure 5.7

Figure 5.1 is the original image marked with 3 ROIs, which are the regions we will be processing in the experiment. Figure 5.3 is the new image with only Intensity histogram stretching. Figure 5.5 is the new image with both Saturation and Intensity histogram stretching. And figure 5.7 is the new image with all Hue, Saturation and Intensity histogram stretching. Figure 5.2, 5.4, 5.6, 5.8 are the corresponding HSI histogram of the left image. And in each histogram figure, first row represents the top left ROI, second row represents the top right ROI, third row represent the bottom ROI.

The core idea of histogram stretching is same as before, and the formula will not be repeated here. The only thing to note is that the range of Hue is (0, 360), the range of Saturation is (0, 100), and the range of the Intensity is (0, 255).

5. Conclusion

Histogram manipulation is an image enhancement method. By comparing Figure 1.1 and Figure 1.3, we can clearly see that after the histogram stretching process, the image is significantly clearer, especially in ROI_1. This is the effect of contrast enhancement. Even as in Figure 2.1 of the General bilinear histogram stretching experiment, we can arbitrarily control the stretching of peak areas or compressing of sparse areas of pixels to achieve enhanced background or enhanced objects effect.

Optimal threshold assumes the image has a bimodal histogram, so objects and backgrounds can be easily distinguished. Therefore, the region with significant bimodal features are ideal for applying this algorithm. In figure 3.2 of experiment 4.3, we can notice the ROI_2 and ROI_3 which has bimodal features get the optimal threshold faster. Another advantage of optimal threshold is that unlike a large computation local operation with adaptive thresholds, the optimal threshold acts as a global operation, resulting in thresholds faster.

As the figure 4.1 in experiment 4.4. The result of using both the Histogram manipulation algorithm and Optimal threshold algorithms is that the background and objects are enhanced at the same time.

In color image histogram manipulation, we can obviously see that intensity histogram stretching enhance the contrast a lot and the effect is remarkable. After continuing the histogram stretching of the S and H components, the effect of continuing the enhancement is small.

In this experiment, there are also some challenges:

- The interval (a, b) that the original image needs to be histogram stretched is defined by user. The most commonly used scheme is generally to choose the smallest and largest grayscale in the image, or the 5^{th} and 95^{th} percentile intensity values in the image. According to different needs, the range we want to enhance the contrast is not the same, it is therefore difficult to automatically manipulate all image histograms based on the same interval criteria.
- Optimal threshold works well in the image with bimodal histogram. But when the image histogram is unimodal or multimodal, optimal threshold is not necessarily the optimal choice.