CAP5400 – Digital Image Processing

Assign 1:

Image Smoothing and Binarization

Name: Zhenglong Wu

Date: Sep. 11th, 2019

1. Introduction

This assignment mainly focuses on the uniform smoothing and adaptive threshold binarization in grayscale images, as well as binarization in color images. And all image processing operations must be performed in the specified non-overlapping region of interest (ROI). There is always noise in any system, and the image is no exception. The smoothing processing of the image can remove the noise very well. Although it also has drawbacks, such as image blurring, etc., we will elaborate in the following sections. Adaptive processing allows the image to be smoothed in a limited space of ROI, no matter how large the adaptive window is, as the window size would gradually reduce when it approaches the ROI boundary. And In the subsequent experiments, the binarization processing performed on the color image was realized by setting a distance threshold. 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. And the two-valued thresholding algorithms used in the previous assignment will not be repeated.

2.1. Uniform smoothing in grayscale images

In the uniform smoothing process of images, there is a sliding window with a user defined size, and the size of the window is odd. Therefore, the pixel on the center of the window can be replaced to a new value, which is the mean of value of the all pixels within the rectangular window. All pixels will be determined by their neighboring pixels, and all weights of the neighborhood is equal. The point of this algorithm is that variance of noise in the average is smaller than variance of the pixel noise. However, this algorithm also has certain drawbacks. For example, when the size of the window is 5x5, its center cannot move to a position that is only 1 pixel away from the ROI boundary, otherwise the window will exceed the ROI boundary. Then many pixels near the boundary will not be processed, the solution to this problem will be explained in the next algorithm.

2.2. Adaptive processing for smoothing window

As described in the previous algorithm, according to the user defined window size, pixels near the boundary cannot obtain information of their neighboring pixels. Therefore, when the window slides to the border, according to the minimum distance (m) from the center of the window to all the boundaries, the window size reduces to (2*m+1)

^2. The idea of the algorithm is that all pixels in ROI other than the pixel located at the ROI boundary position can get an adaptive appropriate smoothing window size to obtain their neighborhood information.

2.3. Binarization for color images

Compared to grayscale images, the color images binarization is more complicated. According to the color principle, all colors are composed of three basic colors, Red, Green and Blue. And their range of color is also 255. Therefore, in each image, every three pixels form a color group. Based on this, there are two more parameters required to be added, a user-defined color and a user-defined distance as a threshold. The pixels within the distance are set to white (255, 255, 255), and the rest to black (0, 0, 0)

3. Description of implementation

The entire codes are developed in the MatLab, there are two sections in the parameter file, one is to set the location (R, C) in top left corner and size (Sr, Sc) of the ROI, the other one is to set all other information of input image, function with arguments and output image. The code reads the parameter file and run the specified functions to get the output image.

parameter.txt – ROI section			
Row	Column	Sr	Sc
28	28	200	200
…	…	…	…

When the code read the ROI_parameter file, it will run a overlapping test to ensure there are no two ROIs overlaps in the same image.

parameter.txt – processing section					
Input	Output	function	Func_Parameter1	Func_Parameter2	…
shore.ppm	Shore_Bi.ppm	colorBi	RGB	Threshold	…
…	…	…	…	…	…

4. Description of results

4.1. Two-valued thresholding within specified ROIs

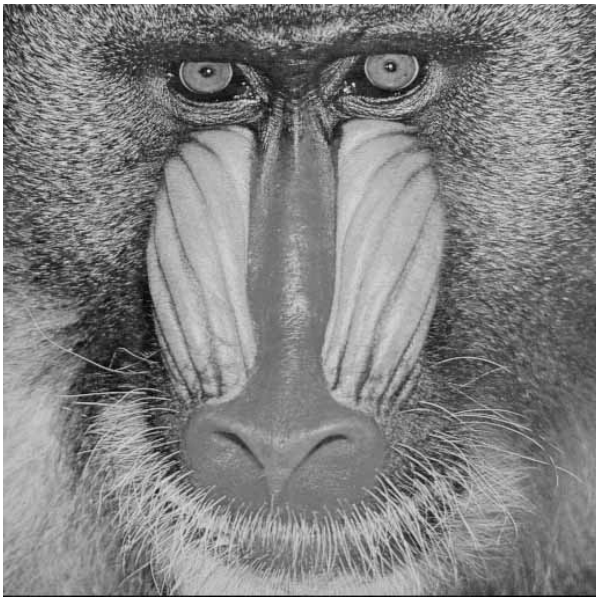

Figure 1.1

Figure 1.2
TwoT1: 50-100; TwoT2: 100-150
TwoT3: 150-200; TwoT4: 100-200

The figure 1.1 is the original image. In Figure 1.2, there are four non-overlapping ROIs with four different two-valued thresholds. The pixels, which is between the two thresholds, are set to “White” and the rest pixels are set to “Black”. The two thresholds of left-top ROI are 50 and 100. Similarly, the right-top ROI are 100 and 150, the left-bottom ROI is 150 and 200, the right-bottom ROI is 100 and 200.

4.2. Uniform smoothing filter in grayscale images

Figure 2.1
Add Normal random noise:
mean=0, sigma=20, 50, 80, 100

Figure 2.2
Uniform smoothing filter 3x3

Figure 2.3
Add Uniform random noise:
20, 50, 80, 100

Figure 2.4
Uniform smoothing filter 3x3

Smoothing filter can reduce the noise in the images. Figure 2.1 and Figure 2.2 is the example pictures that add some noise on the original picture, which based on the model

$$I(x, y) = S(x, y) + n_i$$

In this model, S (x, y) represent the original images or the deterministic signal without any noise, and ni is a random variable, which may distribute in any distribution. Therefore, in this experiment, there are two ways discussed. Figure 2.1 is adding four different noise with a normal distribution in four ROIs. Similarly, Figure 2.3 is adding four different noise with a uniform distribution.

Then Figure 2.2 and Figure 2.4 are the output images after smoothing filter processing on the noise pictures with window size of 3x3.

4.3. Adaptive uniform smoothing filter for larger window size

Figure 3.1
WS: 5x5

Figure 3.2
WS: 7x7

Figure 3.3
WS: 11x11

Figure 3.4
WS: 15x15

In this experiment, based on the Figure 2.1, there are four kinds of window size (WS) that apply to the 4 ROIs with the adaptive smoothing filter algorithm. And the user-defined window size is 5x5, 7x7, 11x11 and 15x15.

4.4. Color image binarization

Figure 4.1

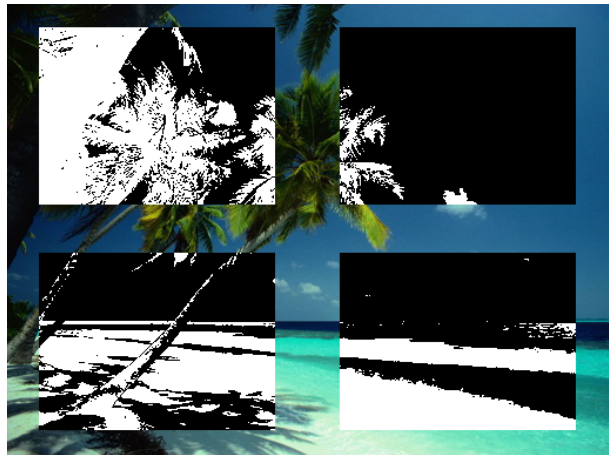

Figure 4.2
User-defined color: RGB (47, 102, 108)
Distance threshold: 5

In the color image binarization, Figure 4.1 is the original shore image. It was composed by three basic color. Then the mean of the Red value is 108 in average calculation, the mean of the Green is 102, and the mean of the Blue is 47.

Therefore, the user-defined color is set by the mean of the Red, Green and Blue to be RGB (47, 102, 108). The distance between original images signal and user-defined color is calculated by Euclidean distance:

$$Dist = \sqrt{\sum_{i=1}^{3} (S_i(\mathrm{x},\mathrm{y}) - C_i)^\wedge 2}$$

S_i (x, y) is a certain signal from the image, C_i is a deterministic user-defined color. And i from 1 to 3 represents red, green and blue.

Figure 4.2 is the output image that set the pixels which has lower distance than 5 to “White” and the rest pixels are set to “Black”.

5. Conclusion

From the experiment 4.2, we can conclude that uniform smoothing filter has effect in reducing the noise in the images, but it also has an obvious drawback, the uniform smoothing filter will blur the image. And according to the comparison between Figure 2.3 and Figure 2.4, uniform smoothing filter works better in the uniform random noise than normal random noise. From the experiment 4.3, we can note that the image become more blurred with WS size increasing.

In this experiment, we can also find some challenges worth to conquer:

- Uniform smoothing filter is not enough to process noise with large fluctuation. And how to try to contain important information of the image with the noise reducing process?
- Due to the larger WS size would return more blurred image, how to find an optimal window size in smoothing filter?
- How to select a user-defined color to get a better performance in color image binarization?