Assignment 1: Threshold and Smoothing Operations

Garrett Moncrief
Dept of Computer Science and Engineering
University of South Florida, Tampa, Florida, USA

1. INTRODUCTION

This assignment focuses on the implementation of image smoothing and the application of threshold manipulation. Especially of interest is applying these modifications to a region of interest (ROI) while leaving the rest of the image unprocessed. This report is organized by giving a description of the algorithms, a description of the implementation, the results obtained, and finally a conclusion.

2. DESCRIPTION OF ALGORITHMS

In this section a description of the algorithms used will be provided.

ROI Selection - Using input parameters of x, y, sx, and sy while the image processes from i0, j0 to in, jn it will apply the changes when it reaches the region of x-sx, y-sy to s+sx, y+sy.

Color Binarization - Using the values of either the red, green, or blue channel the value is compared against a threshold value. The RGB values lower than the threshold become white while those above the threshold value are black.

2D Smoothing - Using an odd window size the smoothing algorithm takes the neighboring cell values and average them by the number of cell values used. For instance, for a window size of 3 the values of i-1,j-1 to i+1,j+1 are taken and divided by 9.

1D Incremental Smoothing - With incremental smoothing the sum of the window size is obtained and then for each increment the oldest value is removed from the sum while the next value is added to it. For instance, for i the sum would be i-1, i, i+1 and then for i+1 the sum would be i, i+1, i+2.

3. IMPLEMENTATION

The code was modified and developed in C++ in Visual Studio on a Windows 10 machine. The program was run directly through Visual Studio, using a parameter file that loaded images from an input directory and saved output files to an output directory.

On the parameter file the user must first note the number of images to be created. They then must specify the input and output paths. Then they must specify the operation and the number of ROI to be included, from 1 to 3. For the ROI the x, y, sx, and sy values must be specified. If more than one ROI is to be implemented the additional parameters must be included before beginning the next ROI.

For color intensity the user must specify dG dR and dB for the channels followed by the modification value. For color binarization the user must specify the color value to be compared against the threshold value.

4. DESCRIPTION AND ANALYSIS OF RESULTS

This section illustrates the results of the algorithms used.

Figure 1. The original image used for RGB intensity function.

```
1
../input/Color/004.ppm ../output/test.ppm
rgbmod 3 150 150 100 150 dR 100 dG 20 dB 20
400 300 50 50 dR 20 dG 20 dB 100
400 100 50 50 dR 20 dG 100 dB 20
```

Figure 2. Output after using the above parameter file input

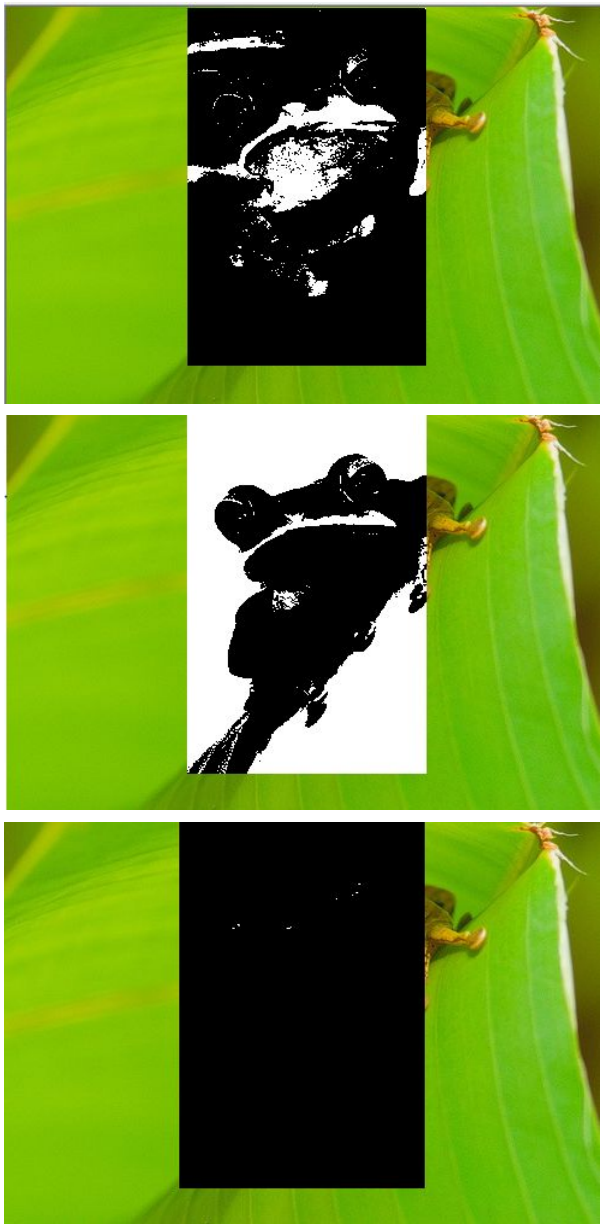
In each ROI a separate value is being emphasized: red, green, and blue. The values entered in the parameter file by the user are added or subtracted to the value of the color in the original image to be output in the target file.

Figure 3. Original image used in color binarization function.

```
3
../input/Color/frog.ppm ../output/test.ppm
rgbbin 1 250 150 100 150 R 150

../input/Color/frog.ppm ../output/test2.ppm
rgbbin 1 250 150 100 150 G 150

../input/Color/frog.ppm ../output/test3.ppm
rgbbin 1 250 150 100 150 B 150
```



Figures 4, 5, 6. Images obtained by applying the same threshold value to the red, green, and blue color values respectively.

In the three files generated a separate color value is being compared against the same threshold value. The most dramatic result is the middle picture which uses the value of green as the comparison value. Since the source image has little blue values in it the third image generates a nearly black result.

Figure 7. Original image used for 2D smoothing.

Figure 8. Image after 2D smoothing

Figure 9. Image used for 1D incremental smoothing.

Figure 10. Image after 1D incremental smoothing.

While subjective, it is my opinion that the image generated by 2D smoothing blends with the untouched image better than the incremental 1D smoothing function. That said, the incremental 1D smoothing function had a quicker runtime than the 2D smoothing function. This would lead to the conclusion that performance is being traded off for speed.

5. CONCLUSIONS

This assignment focused on the different techniques involved with transforming colored image channels and smoothing b&w images, especially as utilized within regions of interest. All of the functions perform in O(N^2) as they iterate through the rows and columns of the source and target images.