198:334 Digital Imaging and Multimedia
Fall 2016

Assignment 3
Due Date – Sunday November 6th at 10pm.

Submission instruction:
- You need to create a report where you put all your results (use MS word or other word processors). Make sure to put all the images you created (whenever the assignment says "print" or "plot" something, it means to include it in the report), and the codes you written in that report.
- Late submission policy: 10% off the grade for each day late.

All the images for this assignment are at Sakai Resources

[Part I: 10pts each]

I.1 – I.4 Solve the following Exercises from the DIP textbook.
First Edition: 6.1, 6.2, 6.6, 7.1
Second Edition: 5.1, 5.2, 5.9, 6.1

I.5 Describe the effects of the linear filters with the following kernels

0	0	0
0	0	1
0	0	0

0	0	0
0	2	0
0	0	0

0	0	1/3
0	1/3	0
1/3	0	0

[Part II:]

[QII-1 - 30pts]
Edge highlights is an effect that accentuates boundaries and edges in images. Unlike sharpening, which increases the contrast around edges, the goal here is to add highlights with certain color at the edges. This can be achieved by adding a scaled version of the magnitude of image gradient to the image. In a sense this is an alpha blending between the image and it's gradient magnitude.

- Implement an ImageJ plugin that takes a gray-level image and produce bright "whitish" highlights on the edges. You need to make sure that the range of any pixel does not exceed 255. Test the code on the image ireland-03gray.tif

- Modify the plugin such that it takes an input colored image and produce the highlights on the edges. You need first to obtain a gray-scale image by averaging the three color channels, and then perform the gradient operation on that image. The scaled magnitude of the image gradient is then added to each of the color channels. Test the code on the provided image. Test the code on the image Amsterdam.JPG

- Modify the plugin such that it produces dark highlights on the edges. Test the code on the image Amsterdam.JPG

Submit the three versions of the plugin (or one version that can deal with the three parts above)

[Q II-2 – 40pts]

Point operations on two images can be used to detect moving objects by subtracting consecutive images in an image sequence. In this problem we will use a sequence of operations to detect soccer player in a field from two frames. Use the images soccer1, soccer2 for this problem. In general we call them images I_1, I_2. Use ImageJ to implement the following steps:

1- Given two images I_1, I_2, use ImageJ point operation to obtain the absolute difference image $d=abs(I_1-I_2)$.
2- Plot the histogram for the absolute difference image d.
3- Visualize the absolute difference image d. To get a good visualization, you need to stretch the range of values in d to be from 0-255. Use a point operation to achieve this stretching. Show the result.
4- Threshold the difference image d by using a suitable threshold a_{th} [Eq 5.5] to obtain a binary image (mask) where the moving pixels is set to a value 1 and the stationary pixels is set to 0. Choose a threshold such that p% of the pixels are above the threshold. Try with p=2%, 1%, 0.1%, 0.01%. In each case show the resulting mask. The mask should show the moving objects in the scene. Which p gives you the best results qualitatively?

5- We can obtain better results by using morphological close operation. Use ImageJ close operation with suitable window size. You can try two different options here

I. Use binary morphological 'close' on the mask m.
II. Alternatively, use gray-level morphological 'close' on the difference image d before thresholding to obtain a better mask

Which of the above methods give you better results? You don't need to show results with different thresholds here. Just show the results with suitable threshold.

6- Use the obtained mask to visualize the moving pixels. Let *m* be the binary mask obtained in step 4. We need to obtain image I_m such that:

$$I_m(u,v) = \begin{cases} I_2(u,v) & \text{if } m(u,v) = 1 \\ 0 & \text{if } m(u,v) = 0; \end{cases}$$

Show the image I_m showing the best result you get using the different options and thresholds from part 4 and 5.

- The same approach can be used in surveillance to detect moving people and vehicles. Re-run your code on images 1015,1020, 1025.jpg to detect the people in a parking lot scene. Show the final result.

[Q II-3 - 25pts]

Point operations on multiple images can be used to add the effect of motion blur given a sequence of images. In this problem we will use alpha blending (page 83) on an image sequence to obtain a motion blur effect. Let $I_1, I_2, \ldots, I_T$ be a sequence of images. We can recursively blend the images into one image B as follows

Initialize B:

$B = I_1$

Recursive blending:

B = alpha B + (1-alpha) I_t , t=1,…,T

Use ImageJ to implement the above blending of multiple images. Use the image sequence 100.tif,…, 110.tif to test your code. Try with different alpha values= 0.1, 0.2, 0.5, 0.8. Show the resulting image in each case.

[Q II-4 – 15pts] Design a linear filter that creates a horizontal blur over a length of 7 pixels, thus simulating the effect of camera movement during exposure. Implement this horizontal blur filter and test your code on the image 100.tif as well as your selfi image from the previous assignment.