COMPUTER SCIENCE 177A (SPRING TERM, 2016)
INTRODUCTION TO SCIENTIFIC COMPUTING

PROGRAMMING ASSIGNMENT 3

For the first three problems, write all functions necessary to solve the problems (that is, don't use `histeq`, `medfilt2`, `imquantize`). For problem 4 you are allowed to use built-in MatLab functions in your implementation.

1. Image Quantization

Write a program that would reduce the number of gray levels in a PGM image from 256 to: 128, 32, 8, and 2. Show your results on the "lena_256" and "peppers_256" images.

2. Median Filtering

- Implement median filtering; the size of the filter should be a parameter. Take the "lena" and "portofino" images and corrupt them with "salt and pepper" noise (i.e., randomly change x% of the pixels values to either black or white; for each image, using x=30 and x=50).
- Apply median filtering to each of the corrupted images. For comparison purposes, apply averaging as well. Try two different mask sizes for each case: 7·7 and 15·15.
- Show and discuss your results for both median filtering and averaging.

3. Histogram Equalization

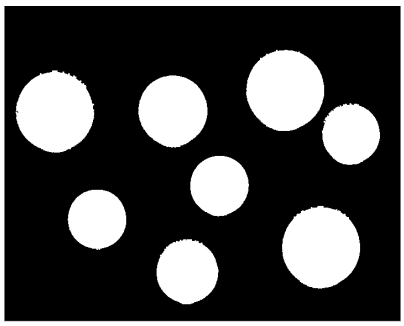
- Write a computer program to compute the histogram of an image.
- Implement the histogram equalization technique. Test your algorithm on a small image (e.g., 5·5) to make sure that it works correctly.
- Perform histogram equalization on the "boat" and "peppers" images.
- Show and discuss your results (i.e., original images/histograms, output images/histograms).

4. Find the coins

Write a program that will process the image coins0.png and count the number of pennies, nickels, dimes, and quarters in the image.

First, change the color picture of the coins into a black and white picture, where the coins are white and the rest is black. (Note that in an image, black is represented with

zero and white with ones. The assumption is made that background data default to zero while ones represent objects in the foreground). To convert a black and white image, use a combination of the built-in MatLab functions: `im2bw`, `imfilter`, and `imfill`. The picture below shows how your image should look like after this process.



The next step is to distinguish between the different coins. The general idea is to be able to calculate the area of each coin. The built-in function `bwlabel` and `bwarea`, should help you with this task.

5.

Write a script that will read from a file x and y data points in the following format:

x	0	y	1
x	1.3	y	2.2
x	2.2	y	6
x	3.4	y	7.4
…			

The format of every line in the file is the letter 'x', a space, the x value, space, the letter 'y', space, and the y value. First, create the data file with a number of lines (that is, do not assume that the number of lines in the file is known) in this format. Do this by using the Editor/Debugger, then File Save As xypts.dat. The script will attempt to open the data file and error- check to make sure it was opened. If so, it uses a loop until the end of the file is reached and fgetl to read each line as a string. In the loop, it creates x and y vectors for the data points. After the loop, it plots these points and attempts to close the file (the number of points, however, should be in the plot title). The script should print whether or not the file was successfully closed.

Submission:

- Your code should be submitted electronically via Latte the day it is due, **Tuesday, March 15 at 11:55pm**.
- Save all your files in a directory named HW3_LastName_FirstName and zip it. Upload the zip file onto Latte.
- Please name each file **q1.m**, **q2.m**, etc. except if the name of the function or the script is specified in the problem.
- For the late policy check Lecture 1.