The Hong Kong University of Science and Technology
Department of Computer Science and Engineering
COMP4421 (Fall 2016)

Assignment 1

Total = 100 marks
Due: 11:55pm, Oct. 17, 2016
Assignments must be submitted via Canvas
Late Policy: 10% reduction; only one day late is allowed, i.e., 11:55pm, Oct. 18.

Overview

In this programming assignment, you will use MATLAB to create one gradient magnitude image, to program an additive (zero mean) Gaussian noise generator, arithmetic mean filter, Wiener filter, high frequency emphasis filter, and alpha-trimmed mean filter. A set of M-files can be obtained from the course website. The routine found in the comp4421_assign1.m file performs a series of image processing and display operations on a pre-defined grayscale image. You are asked to complete the missing implementations of functions in the programming section.

This programming assignment should be submitted via the Canvas system on or before the due date.

Programming assignment specifics (100%)

Part 1: Create one gradient magnitude image from a grayscale image. (10%)

You need to complete the function in the file "grad_mag_image.m". The function "grad_mag_image" takes a grayscale image of type uint8 as input and returns a gradient magnitude image of the same data type. $mag(\nabla f)$ at z_5 can be calculated with the following equation:

$$mag(\nabla f) \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

z1	z2	z3
z4	z5	z6
z7	z8	z9

Part 2: An additive Gaussian noise generator. (15%)

You need to complete the implementation of an additive Gaussian noise generator in the gen_gauss_noise.m file. The routine gen_gauss_noise takes three numbers as input: size of an image along the X-axis and Y-axis, and the standard deviation of the Gaussian noise. You should generate an image of additive Gaussian noise with the given standard deviation. The data type of the output image should be double.

Hint: You can find a MATLAB internal function to generate random numbers with Normal distribution.

Part 3: Arithmetic mean filter. (15%)

You need to complete the implementation of the routine in the arithmetic_mean_filter.m file. The routine takes a noisy image (in grayscale format of type uint8) as input and attempts to remove the noise with an arithmetic mean filter. A 3×3 window is used to filter the noisy image. The output image type should be uint8.

Part 4: Wiener filtering with the power spectra of noise and un-degraded image. (15%)

You need to complete the Wiener filter, suppose we know the power spectra of the noise S_η and the un-degraded image S_f. You need to complete the implementation in the wiener_filter_1.m file. The data type of the output image should be uint8. Do not use the internal MATLAB function "wiener2" for implementation.

Part 5: Wiener filtering with a constant K. (15%)

You need to complete the Wiener filter, suppose we DO NOT know the power spectra of the noise S_η and the un-degraded image S_f. This routine employs the degradation function H_d (given in the main routine) and a constant $K \approx S_\eta / S_f$ to filter the given noisy image. You need to complete the implementation in the wiener_filter_2.m file. The data type of the output image should be uint8. Do not use the internal MATLAB function "wiener2" for implementation.

Part 6: High frequency emphasis filter. (15%)

You need to complete the high_freq_emphasis.m file. The function emphasizes the high frequency components of an image in the frequency domain. You also need to complete to the Butterworth filter in ButterWorth.m as a step of implementing the high frequency emphasis one. Different values of a and b will be applied to the filter (see comp4421_assign1.m part 6). The output image type should be uint8.

Part 7: Alpha-trimmed mean filter. (15%)

You need to complete the atrimmed_mean_filter.m file. The routine takes a noisy image (in grayscale format of type uint8) as input and attempts to remove the noise with an alpha-trimmed mean filter. Different values of d will be applied to the filter (see comp4421_assign1.m part 7). A 3x3 window is used to filter the noisy image. The output image type should be uint8.

Sample run of the programming assignment

The main routine of the assignment including displaying the final results is well written in the comp4421_assign1.m file. After completing all the functions, you are supposed to get three figure on the screen when you run the command below in the MATLAB environment. The three figures are the results of part 1 to 5, part 6 and part 7 respectively.

```
>> comp4421_assign1;
```