EE5806 Topic in Digital Image Processing

(Due date for Part I and II: 6 Apr. 2016)

Part I. Homework Assignment

1. The histogram of an image is given in Figure 1 below.
   a. Design a 2-level quantizer to quantize the pixel values of the image to only 2 levels. Mark clearly on the histogram below the decision levels (using t_1) and reconstruction levels (using r_1 and r_2) on their likely locations. Find the optimal values for t_1, r_1 and r_2 that will minimize the Mean-Square-Error (MSE).
   b. Determine the MSE of the resulting quantizer.

   Assume the image can take on any continuous value between 0-256.

(20 marks)

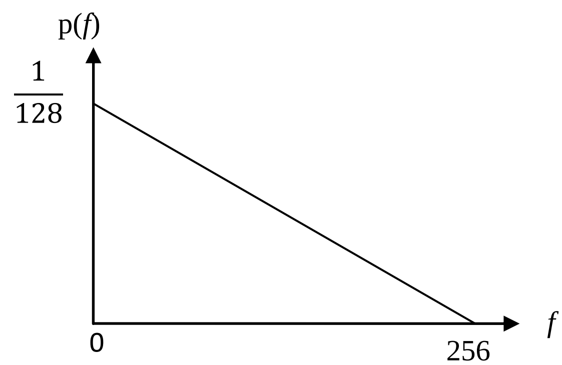

Figure 1

2. The histograms of 3 images are given in Figure 2 below.
   a. Sketch an appropriate transformation function for each image that will likely improve the contrast of each image. The resulting image pixel value should have a range of 0-255. Mark any transition points on your transformation functions clearly.

(15 marks)

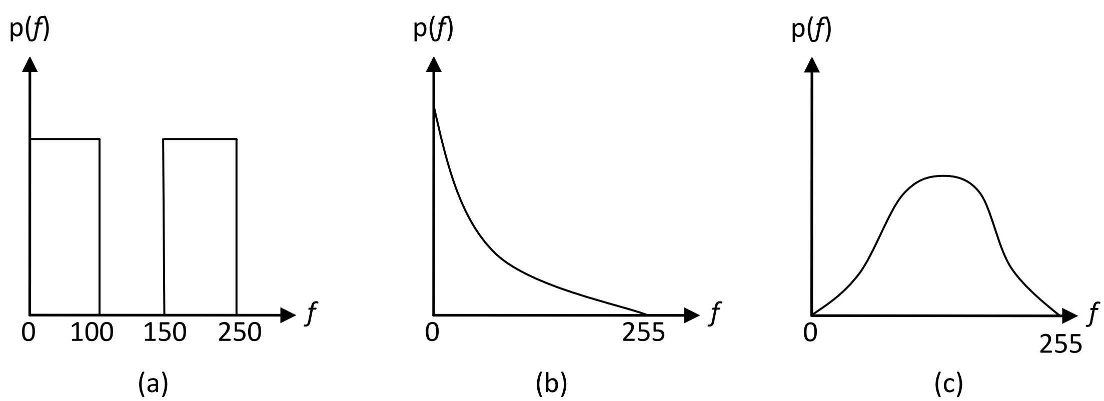

Figure 2

3. For the image shown below, find a transformation function that will approximately equalize its histogram, and draw the transformed image. Give the histogram of the original image, the histogram of the processed image, and the transformation function (in a lookup table). Assume that the processed images can only take integer values between 0 and 7 (including 0 and 7).

(20 marks)

0	1	1	3	4
0	2	3	4	4
2	3	4	4	5
3	4	4	5	6
4	4	5	6	7

4. For each filter given below, answer the following questions,
   a) Is it a separable filter? If yes, present the horizontal and vertical filters, and
   b) What is the functionality of the three filters? Explain your reasoning.
   c) For the H_1 filter below, determine the Discrete-time Fourier Transform (DTFT) $H(u, v)$ of the filter (assuming the origin is at the center), and sketch the one dimensional profiles $H(u, 0)$ and $H(0, v)$.
   d) What is the function of this filter based on its frequency response?

(20 marks)

$$H_1 = \frac{1}{8}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 12 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad H_2 = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad H_3 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

5. Consider an edge detection algorithm using two filters given below. Specifically, the given image is filtered by the each filer H_x and H_y to obtain G_x and G_y, and then a gradient magnitude is obtained by using $G_m = |G_x|+|G_y|$. If the gradient magnitude is greater than a threshold T at any pixel, this pixel will be considered as an edge pixel.
   Using the above algorithm, for the sample image F given below, determine the Gx and Gy image, the edge map E (which has a value 255 if a pixel is not an edge, and has a value of 0 if a pixel is an edge) for the sample image given below and set T=25. When filtering, perform symmetric extension on F to determine the values outside the boundary. Only G_x image, G_y image, and edge map within the boundary of the given image is needed to show.
   Note that the origin of the filters H_x and H_y are both assuming at the top-left corner.

(25 marks)

$$F = \begin{bmatrix} 100 & 100 & 0 & 0 \\ 100 & 100 & 0 & 0 \\ 100 & 100 & 0 & 0 \\ 100 & 100 & 100 & 100 \end{bmatrix}, \quad H_x = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \quad H_y = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

EE5806 Digital Image Processing

Part II. Computer Project: Image Processing in Spatial Domain

In this project, you need to implement image processing methods in spatial domain.
You can download and use the input images as follows:

• For Parts (1) and (3), use “picture_*n*.bmp”, where “*n*” stands for the last digit of your student ID. For example, if your ID is 67812345, use “picture_5.bmp”. If your ID is 32106789, then use “picture_9.bmp”.

• For Parts (2) and (4), use “picture_1+*n*.bmp”, where “1+*n*” stands for the last digit from your student ID plus one. For example, if your ID is 67812345, use “picture_6.bmp”. If your ID is 32106789, then use “picture_0.bmp”.

The image file assignment table is shown in Appendix. You should write computer programs (either C or Matlab [without using any Toolbox such as Image Processing Toolbox function]) to implement the functions specified below.

1. **Histogram processing.** Apply the Gamma transformation to “picture_*n*.bmp” with γ=0.4 and γ=2 (c=1). Compute the intensity histogram of the original and transformed images, and then perform the histogram equalization on each of them. Your output should contain 12 diagrams, including one original and two transformed images and their corresponding histograms, and three equalized images and their corresponding histograms.

(20 marks)

2. **Bit plane slicing.** Decompose “picture_1+*n*.bmp” into its bit planes, and then reconstruct the image using the two most significant bit planes. Compute the histogram by setting the least and most significant bits of each pixel of “picture_1+*n*.bmp” to zero respectively. Your output should contain 8 diagrams, including the original and reconstructed images, the binary images of the least significant bit (LSB) and the most significant bit (MSB) planes, and the histograms and modified images by setting LSB=0 and MSB=0.

(30 marks)

3. **Low pass filtering for noise removal.** Add the salt and pepper noise to “picture_*n*.bmp”. Apply the low pass filtering and median filtering masks of sizes 3×3 and 7×7 to the corrupted image respectively. Your output image should contain 6 diagrams, including the original and corrupted images, the blurred images with two filtering masks of two sizes.

(30 marks)

4. The Laplacian filtering for image sharpening. Blur the image of "picture_1+*n*.bmp" with low pass filtering masks of size 3×3, and then use the two discrete Laplacian masks with and without elements in two diagonal directions to filter the blurred image. Your output should contain 6 diagrams, including the original and blurred images, two Laplacian images and two Laplacian-filtered images.

(20 marks)

Appendix:

The last digit of your student ID, n	Part 1 image file	Part 2 image file	Part 3 image file	Part 4 image file
0	"picture_0.bmp"	"picture_1.bmp"	"picture_0.bmp"	"picture_1.bmp"
1	"picture_1.bmp"	"picture_2.bmp"	"picture_1.bmp"	"picture_2.bmp"
2	"picture_2.bmp"	"picture_3.bmp"	"picture_2.bmp"	"picture_3.bmp"
3	"picture_3.bmp"	"picture_4.bmp"	"picture_3.bmp"	"picture_4.bmp"
4	"picture_4.bmp"	"picture_5.bmp"	"picture_4.bmp"	"picture_5.bmp"
5	"picture_5.bmp"	"picture_6.bmp"	"picture_5.bmp"	"picture_6.bmp"
6	"picture_6.bmp"	"picture_7.bmp"	"picture_6.bmp"	"picture_7.bmp"
7	"picture_7.bmp"	"picture_8.bmp"	"picture_7.bmp"	"picture_8.bmp"
8	"picture_8.bmp"	"picture_9.bmp"	"picture_8.bmp"	"picture_9.bmp"
9	"picture_9.bmp"	"picture_0.bmp"	"picture_9.bmp"	"picture_0.bmp"