Digital Image Processing

Instructor: Mohsen Gharibnaseri

Assignment 2

Intensity Transformations and Spatial Filtering

Due Date:
25th of june 2021

Mohsen Gharibnaseri
MOHSEN.GHNASERI@YAHOO.COM

Problem 1

The focus of this problem is to experiment with intensity transformations:

a. Load the image *"spine.tif"* and apply a suitable gamma transformation to enhance the contrast of the image. Use *"subplot"* function to display the resultant images with different parameter values and their histograms using *"imhist"* function.
b. Repeat part (a) by applying a suitable log transformation to enhance the contrast of the original image used before.
c. Load the image *"kidney.tif"* and implement the transformations given in Figure 1 and Figure 2. Display the transformed images and comment on their quality. The range of the highlighted interval for Figure 1 is [A, B] = [160, 255] and the intensity values are $s_L = 10$ and $s_H = 250$. For the case of Figure 2, the range is [A, B] = [80, 160] while the intensity is set to $s_L = 15$.
d. Implement a program to perform bit plane slicing. Extract and display the resulting separate plane images from MSB to LSB. You should use the image *"dollar.tif"* and the *"subplot"* function to display the resultant images.
Hint: You can use the "bitget" function.

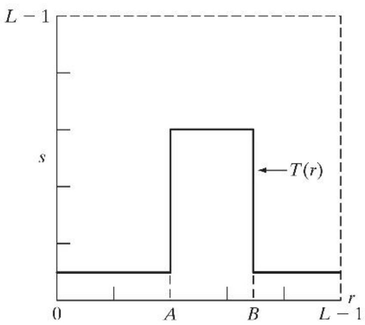

Figure 1

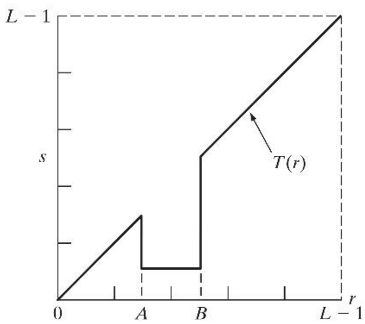

Figure 2

Problem 2

In this problem, we want to be acquainted with histogram processing to enhance the visibility and contrast of images. As you know, visibility of objects, road signs, and many other things is reduced at night. Therefore, you can use histogram equalization to overcome the issues.

a. Load the images *"road1.jpg"*, *"road2.jpg"* and *"road3.jpg"*. Compute their histograms using your own code. You are not allowed to use *"imhist"* function.
b. Apply the global histogram equalization on the images and display histograms before and after equalization with the corresponding images.
c. Now, apply the local histogram equalization on the images using various block sizes to reach an optimal result. Display histograms before and after equalization with the corresponding images.

d. Repeat part (b) and part (c) on the image *“squares.tif”* and display all the results.
e. Discuss the results of global and local histogram equalizations.

You are not allowed to use MATLAB built-in functions.

Problem 3

Sometimes we need to approximate the histogram of an image based on another target image. Therefore, we use histogram specification. Load the images *“ImageOrg.png”* and *“ImageGoal.png”*. Then:

a. Generate an algorithm to modify histogram of *“ImageOrg.png”* in such a way that the outcome nearly approximates the histogram of “*ImageGoal.png*”. Display and discuss the original image/histogram, specified histogram, output image/histogram, and then compare and discuss them. Name the output image as *“spec”*.
b. Repeat part (a) using local histogram equalization instead of global equalization with an appropriate block size. Compare and discuss the results of part (a) and part (b).
c. Now, we want to return the histogram of *“spec”* in previous part to the original histogram (histogram of *“ImageOrg.png”*). Please do the desire, display output image/histogram, and original image/histogram. If there are differences between outputs, original image, and their histograms, explain the reasons.

You are not allowed to use MATLAB built-in functions.

Problem 4

In this problem, we investigate the sharpening filter described in Section 3.6.3 of the textbook, *unsharp masking and highboost filtering*.

a. Implement the convolution operator for applying spatial filter mask on the image.
b. Load the image *“moon.tif”* and blur that by using smoothing filter masks shown in Figure 3.32 of the textbook.
c. Subtract the blurred images from the original image.
d. Add the results of the former part with various weights to the original image.

You are not allowed to use MATLAB built-in functions.

Problem 5

Most of the time, it is common to combine different spatial filtering methods to enhance the quality of an image. According to Section 3.7 of the textbook:

Load the image *“bone-scan.tif”*. Then:

a. Compute Laplacian of the image, using Figure 3.37 kernel.
b. Try to get a sharpened image by adding the result of part (a) to the Input.

c. Compute Sobel gradient of input using Eqs. 3.6-12 of the textbook.
d. Smooth the result of part (c) using a 5×5 averaging filter.
e. Generate a mask image using the product of part (b) and part (d) results.
f. Add the result of part (e) to the input.
g. Get the final image by applying a gamma transformation with an appropriate parameter to the result of part (f). Display all the results.

You are not allowed to use MATLAB built-in functions.
