Digital Image Processing

Instructor:Mohsen Gharibnaseri

Assignment 1

Digital Image Fundamentals

Due Date:
29th of June 2021

Mohsen Gharibnaseri

MOHSEN.GHNASERI@YAHOO.COM

Problem 1

Here, we have an example to get familiar with MATLAB for Digital Image Processing.

a. Use *imread* function to read image “lenna_rgb.png.”
b. Use *imshow* function to show the image.
c. Now, convert the image to grayscale using *rgb2gray* function and display the result.
d. Convert the gray image you made in step “c” into double using *im2double* function and Discuss the differences.
e. Save the image you made in step “d” in “.jpg” format.
f. Magnify and display the gray image by scales 5, 1/2, and 1/4, then discuss the effect of reducing spatial resolution.
g. Like Figure 2.16 in the textbook, try to plot the scan line of 150^{th} line of the gray image.

Problem 2

Now load "skull.tif" image. As you see, this is a 256-intensity level image. Display the image in 64, 16, 4, and 2 intensity levels while keeping the image size constant.

Discuss the effect of varying the number of intensity levels in a digital image.

Problem 3

As mentioned in the textbook, geometric spatial transformations and image registration are necessary steps for some image processing applications. In this problem, we will try to get to know some transformations used in this field.

Load the image “T.tif” and apply all the transformations such as Scaling, Rotation, Translation, and Shear to it. You can use the Affine Matrices mentioned in Table 2.2 of the textbook. Set the parameters of each transformation by your own idea and report them clearly for each part. Display the results.

Problem 4

In this section, we want to implement some important logical operations.

a. Create two arrays of zeroes 200×200.

b. In the first array, add a rectangle of amplitude 1 starting from (20,20) with a length of 100 and a height of 80.

c. In the second array, add a rectangle of amplitude 1 starting from (70,40) with a length of 110 and a height of 120.

Write functions to implement logical operations brought in Section 2.6.4 in the textbook (NOT, AND, OR, AND-NOT, XOR), just like Figure 2.33. The functions must get the name of each operation. Display your results like Figure 2.33.

Problem 5

Local averaging is a simple, intuitive, and easy way for smoothing images and is often used to reduce noise. The idea of local averaging is to replace each pixel value in an image with the mean value of its neighbors, including itself, like Figure 2.35 of the textbook.

a. Write your own code to do local averaging on "lenna.png" with a window size of 3×3.

b. Repeat "Part a" with different window sizes such as $7 \times 7, 11 \times 11$, and 23×23. Discuss the effect of window size on local averaging.

Problem 6

An important application of image multiplication (and division) is *shading correction*.

a. An imaging sensor is used to image a *tungsten filament*. Load the image "filament.tif."

b. Suppose you have the *shading function*. Load the image "shading.tif."

c. Eliminate the shading pattern from the image using the method discussed in the textbook.

d. Suppose that the shading function is unknown, but you have access to the imaging system. What will you do to eliminate the shading pattern?

Problem 7

Interpolation is the process of using known data to estimate values at unknown locations. In this problem, we want to compare three different interpolation approaches used for image shrinking.

a. Load the image "watch.tif."
b. Shrink the images by a factor of 5 (you can use *imresize* function) using *nearest*, *bilinear*, and *bicubic* interpolation techniques. Zoom the reduced image back to its original size and discuss their differences in terms of details.