Digital Content Retrieval - Exercises

VERY IMPORTANT: Please assure that you have a valid login and password for the PC's of room C2 and C3. If not, register yourself giving your surname, name and mat code at the technicians of room C4 (ref: Mrs.Turri)

1. Download from the net a grey level image in the field of visible, discover the resolution (n. of pixels) , and compute the space necessary to store it, by assuming to code it in the pure binary number code. Read the image, diplay it with a title, and compute the space requirements by Matlab. Help: Feel confident and try the basic following Matlab built in functions: input, imread, figure, title, xlabel, imshow, clear, sizeof.

2. Write the code capable of reducing the number of intensity levels in a image from 256 to 2, in integer powers of 2. The desired number of intensity levels needs to be a variable input to your program.

3. Write the code to add Gaussian noise to an original, preferred image. You must be able to specify the noise mean and variance. Write the code to add salt-and-pepper (impulse) noise to an image. You must be able to specify the probabilities of each of the two noise components.

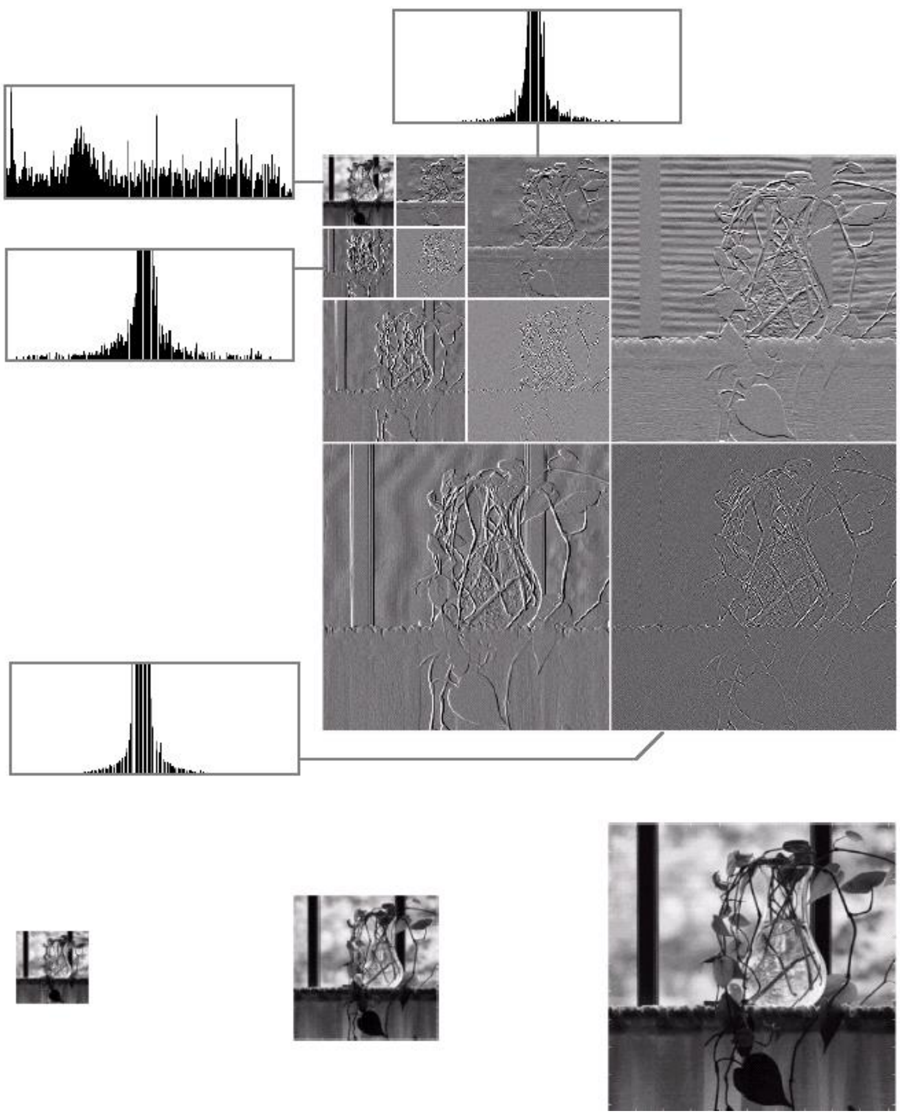
4. Write a program to compute the root-mean-square error of a compressed- decompressed image.

5. Download an image (if color image, perform a conversion to a gray level image) and write the code which computes the entropy. Compare the computed value to the value provided by the built-in matlab function entropy

6. In order to understand the concept of non-stationarity of the image signal (see for reference to figure 7.1 of the book,) write the code which extracts 4 blocks of 32 X 32 pixels of a preferred image and compute and plot the histograms.

7. Download a gray-level image (or convert a color image to a grey-level one) compute the discrete cosine transform (block 8x8) and plot it at a proper scale. Threshold the values of the coefficients of the cosine transform (the threshold is parametric) and perform the inverse discrete cosine transform. Try several values of threshold (i.e., the mean value of the coefficients, the maximum value/2, …)
8. Compute the root mean squared error between the original image of point 3 and the noisy impaired versions created as outputs of point 3.

9. With reference to figure 7.8 of the book, write the code to extract 4 bands of the wavelet and compute the histogram on a preferred image.
10. Write the code for computing the Haar Wavelet and use it to extract the horizontal edges (hint: choose an image where horizontal edges are evident and predominant).
11. Write a code for adding Gaussian noise to an image and use the Haar wavelet for denoising it.
12. Write the code for computing one of the Daubechies Wavelet (choose the one you like), discard a certain amount of coefficients and perform the Inverse transform. Compute the root mean squared error between the original image and the reconstructed one.
13. Plot the value of SSIM index (Y-axis) vs. the quality factor (X-axis), for ten versions of a lossy compressed jpeg image (chose whatever image you like), by varying the quality factor of JPEG form 10 to 100, of step of 10 (Check if in your current version of Matlab or alternative programming environment there is a built-in function named `ssim`).



a
b c d

FIGURE 7.8 (a) A discrete wavelet transform using Haar basis functions. Its local histogram variations are also shown; (b)–(d) Several different approximations (64 × 64, 128 × 128, and 256 × 256) that can be obtained from (a).