Digital Content Retrieval – Exercises

DIGITAL CONTENT: IMAGES

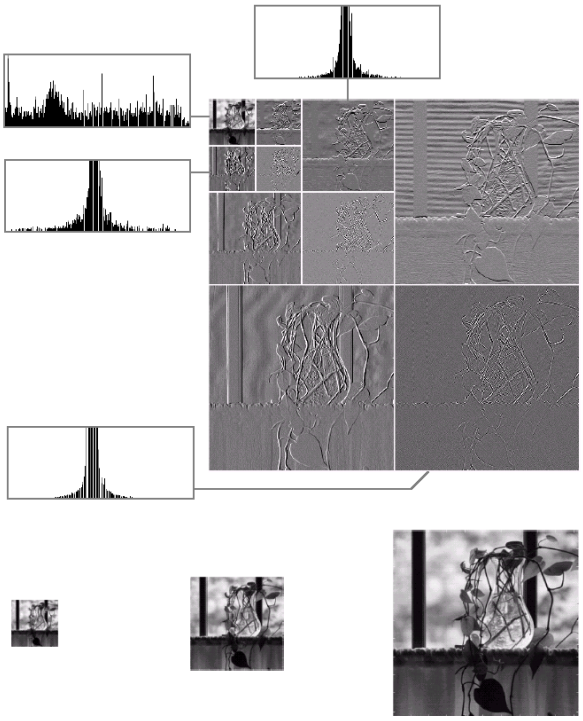
1. Create a folder on your space, open MATLAB and add the folder in MATLAB default path.
2. Download from Internet a grey level image in the field of visible, discover the resolution (n. of pixels), and compute the space necessary to store it, by assuming to code it in the pure binary number code. Read the image, display 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, whos, figure, title, xlabel, imshow, clear, sizeof. Write an .m fie for the execution of the commands in sequence and save it in your working folder. The .m file has to be properly commented. Understand the difference between script and user-defined function.
3. Convert your image in one of the other file format supported by Matlab and write it on disk. Compare the dimensions of the files and try to explain the difference, if any. (hint: see function imwrite()).
4. Select one pixel of the image by asking the coordinate by a keyboard input, modify it in another value (as you like) and display the new image. To see the difference, use the magnifier (zoom) tool.
5. Write the code to compute the histogram of a grey level image (two channel) and display in the same window the image and its histogram. Try the same on a color RGB image, by displaying the histograms on the three-color channel.
6. Perform a histogram equalization to improve the visual quality of the image (hints: try function histeq). Display the original image and the improved image.
7. 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.
8. Write the code to add Gaussian noise to an original, preferred image. You must be able to specify the noise mean and variance of Gaussian distribution as input parameters. Write the code to add salt-and-pepper (impulse) noise to an image. You must be able to specify the density of salt-and-pepper noise as input parameter. Display the original and noisy images.
9. Write a program to compute the root-mean-square error of a compressed- decompressed image.
10. 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.
11. 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.

12. 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, …) to investigate different strategy of thresholding masking.
13. 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 using a zonal masking approach and perform the inverse discrete cosine transform. Try several pattern of zonal masking in the 8X8 matrix to investigate different strategy of zonal masking.

Hints: example of Zonal masking for an 4X4 block:

1111
1110
1100
1000

14. Compute the root mean squared error between the original image of point 3 and the noisy impaired versions created as outputs of point 3
15. 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.



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).

16. 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).
17. Write a code for adding Gaussian noise to an image and use the Haar wavelet for denoising it.
18. 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.
19. Write the code for computing two different Wavelets (one level decomposition) and compare visually the histograms of the different bands.

DIGITAL CONTENT: AUDIO

20. Download or register an audio file in one of the format which can be taken as input by MATLAB. Sound the file at different sampling rate.
21. Perform a dct on the audio signal, plot the coefficient.
22. Threshold the dct coefficient trying different thresholding schemes and recover the signal. Listen to the signal to understand the effects.
23. Perform a wavelet transform on the audio signal, plot the coefficient.
24. Threshold the wavelet coefficients trying different thresholding schemes and recover the signal. Listen to the signal to understand the effects. (Hints: try as first experiment, to discard a complete band
25. Try to use dct or wavelet to denoise the audio signal. If the audio signal is not noisy (i.e., it is not the result of a recording) add a noise to the original signal
26. Plot the Fourier Transform of the signal and compare the different representation of the coefficient with the dct and wavelet transform.
27. Plot the spectrogram of the signal
28. Case study: the voice. Record the same text read by two different voices and compare the spectrogram or the coefficient plots in some transform domain (Fourier, DCT or Wavelet). Where are the differences?

QUALITY OF IMAGES

29. Choose a gray level image and create ten compressed version using jpeg format (hint: use imwrite function with the format jpeg and increasing quality factor, from 10 to 100). Then compute the psnr (between the original image and the compressed versions) and plot it versus the image quality of jpeg.
30. Choose a gray level image and create ten compressed version using jpeg format (hint: use imwrite function with the format jpeg and increasing quality factor, from 10 to 100). Then compute the ssim values (between the original image and the compressed versions) and plot it versus the image quality of jpeg. Repeat the exercise with JPEG2000 format

31. Plot on the same figure the psnr and the ssim values computed between the original image and ten versions of the same image corrupted by Gaussian noise of increasing energy.
32. Do the same of exercise 3 on a color image by computing the psnr and the ssim values separately on the three components and comment the results. Is there a difference among the metrics in the three color components?

COLOR IMAGES:

33. You are requested to create three 3 X3 "images" having only 0 and 1 as elements. These images will be the red, green, and blue component images of a 3X3 color image that you will display. The color image should look like this:

black	red	blue
green	yellow	magenta
cyan	white	white

For example, the red image could be created by the statement
>> r=[0,1,0;0,1,1;0,1,1];
After you have created the g and b images, use the Matlab statements
>> rgb_image=cat(3,r,g,b);
>> imshow(rgb_image);
to display the image. Look at the image carefully to confirm that you have placed the saturated primary colors RGB and CMY in the correct places.
The resulting image plot is rather small (3 x 3 pixels). Use the magnification feature
imshow(rgb_image,'InitialMagnification','fit');
of the imshow command to scale the color image to a larger size for viewing.

34. Scale each color component r,g,b by a constant that is less than one and display it; e.g., scaling by 0.8 is implemented as:

```
>> c=0.8;
>> rgb_scaled=cat(3,r*c,g*c,b*c);
```

Use values of c = 0.8, 0.6, 0.4, 0.2 to observe the effect. What happens to the colors as c is varied over the range of values?

35. Convert the rgb_image to an indexed image using a “proper” number of colors N. Plot the map and check that the colors are correctly mapped. How much is the minimum value of N?

36. Read a jpeg color image and repeat the exercise 35. Discuss how you can vary N.

WATERMARKING

37. Choose a gray level image (asset image). Choose a message image (message). The two images must e of the same dimensions. Extract the 8 bitplane of each image. Hints: use the function bitget, for example for the LSB bitplane

>>B1 = bitget(pic,1)*2^0;

Substitute the first N LSBs bitplanes of the asset with the corresponding bitplanes of the message and display the watermarked image. Try different values of N. At which value of N the watermark becomes visible?

38. Choose a gray level image as asset image. Generate a random Gaussian spread spectrum noise to use as watermark signal of length 1000 (hint: use the function randn(1,1000)). Use the 1000 largest coefficients to embed a watermark sequence of length 1000. The only exception is the DC term, located in (0,0) of the DCT matrix, that should not be changed due to its perceptible change in the whole brightness of the picture. Coefficients are modified according to the stream bits of the message using to the equation:

$$C_{AW} = C_A \cdot (1 + \alpha \cdot W_i)$$

In which CAW is the watermarked coefficient, CA is the original one, α represents watermarking strength (e.g. 0.1), and Wi is the corresponding bit of the message data. Perform the inverse DCT transform to display the watermarked image. Extract the watermark. Compute also the difference between the original watermark and the extracted one.

39. Choose one of the exercise n. 8-38, try, write the m.file with help functionality and detailed comments. Try to add a new functionality to the code (be creative!) and describe the results. (TO BE EVALUATED for mini-project MATLAB, only for face-to-face attending students)