School of Engineering – University of Patras

Computer Engineering and Informatics Department

Signal Processing and Communications Lab

Academic Year: 2021/22 (Spring Semester)

DIGITAL IMAGE PROCESSING AND ANALYSIS

LAB ASSIGNMENTS

PRACTICAL ISSUES:

The report must be delivered in electronic form until May 29, 2022. The electronic deliverables consist of a technical report and the Matlab (or Python) codes you developed. The report should include a brief description of each technique, the results which you obtained in any case and a relevant discussion.

Each student must deliver her/his own report.

We suggest using the latest versions of Matlab (or Python) with the necessary signal and image processing toolboxes.

If you have any questions you may contact:

Prof. Kostas Berberidis (berberid@ceid.upatras.gr) or Mr. Nikos Piperigkos (piperigkos@ceid.upatras.gr).

Requested Tasks:

1. Filtering in the frequency domain

For image **moon.jpg** do:

a. *Preprocessing:* You must linearly transform the range of values in any of the above images, so as to span the available dynamic range [0:255]. The resulting images are the ones that will be processed in the sequel. The above transform must be applied as well before depicting any final result, so that the whole range of the 256 levels is used. Then, using discrete Fourier transform (DFT) properties, make an appropriate shift of the coefficients of the transform, so that the frequency point (0,0) is moved to the center.
b. Compute the two-dimensional DFT with the lines-columns method using the one-dimensional FFT. The amplitude of the transform should be depicted in linear and logarithmic scale.
c. Filter the image in frequency domain (u, v) using a low-pass filter $H(u, v)$ of passband of your own choice.
d. Apply 2D inverse DFT (IDFT) to go back to spatial domain.
e. Follow the inverse procedure of Step 1a. to recover point (0,0).

Notice: You are allowed to use only the MATLAB (or Python) function of 1D-DFT for implementing 2D DFT and IDFT.

2. Image compression using DCT

For image **barbara.mat** do:

In the beginning, divide the image to non-overlapping sub-images of size 32x32. For each sub-image, compute the two-dimensional discrete cosine transform (DCT) and order the coefficients of the transform. From the 32x32 coefficients, keep only a percentage of p coefficients by using:

- Zone method
- Threshold method

Reconstruct each 32x32 sub-image by computing the inverse DCT and finally reconstruct the whole initial image. Give the curve of the mean square error between the initial image and the reconstructed image for p = [5% - 50%}. Depict the results and provide relevant discussion.

Bonus: If you are going to use HSI model, apply the necessary processing in H(ue) and S(aturation) channels and provide discussion for any interesting visual changes.

3. Noise filtering

a. Add to image **tiger.mat** white Gaussian noise of zero mean and covariance corresponding to SNR equal to $15\ dB$. Use moving average and median filters to remove noise from image.
b. Add to image **tiger.mat** impulse noise of 20% rate and use the same filters for noise removal.
c. Add to image **tiger.mat** i) white Gaussian noise of zero mean and covariance corresponding to SNR equal to $15\ dB$, ii) impulse noise of 20%. Successively use both filters in your order of preference for noise removal.

Explain the results of applying the filters to the image according to the type of noise.

4. Shape description

For image **leaf.jpg** do:

a. Write a program which computes the Fourier Descriptors (FDs) of input image.
b. Reconstruct the shapes of the image using as input the FDs. Experiment yourselves with different number of FDs.
c. Repeat the previous tasks for the following geometric transformations of input image: i) $90°$ rotation, ii) 0.5 scaling, and iii) translation by $(40,50)$.

Depict the results and provide relevant discussion.

5. Image reconstruction-deconvolution

For image **lenna.jpg** do:

Part A

Add white Gaussian noise of zero mean and covariance corresponding to SNR equal to $10\ dB$. Using Wiener filter, remove image noise by each of the following two ways:

a. Assuming the power of noise is known.
b. Assuming the power of noise is unknown.

Comment and compare your results in both cases.

Part B

Apply the transformation at **psf.m** file, which implements the point spread function of an unknown image capturing system. Use the command $Y = psf(X)$, where X and Y are the input and output images.

a. Compute the impulse response of the unknown system using appropriate technique and depict its frequency response.
b. To reduce blurring, use the inverse filter at frequency domain with threshold. Depict mean squared error between input and reconstructed image for different values of threshold.
c. Comment when no threshold is used.

6. Image retrieval

You are given the image database **Database.rar** and testing database **test**. For a given input image from **test** folder, you need to efficiently retrieve it from the image database.

a. Retrieval based on **histograms**. Compute the histogram for each image in **Database.rar** and **test**. As output, you have to find the image from **Database.rar** with the most similar histogram of that of query image from **test**. Extract the ratio of successful retrieval.
b. Retrieval based on **DCT**. Compute DCT coefficients for the images of **Database.rar** and **test** in order to compare the images in DCT domain. Using threshold method, keep only 10, 50 and 100% of DCT coefficients. Extract the ratio of successful retrieval.

In all cases, use mean squared error as the similarity metric either for histograms or DCT.

Literature

[1] “Digital Image Processing”, Rafael C. Gonzalez, Richard E. Woods, Prentice Hall, Inc. (Editions 3nd or 4th).

[2] “Mathematics of Digital Images”, S.G. Hoggar, Cambridge, 2006.