Cairo University- Faculty of Engineering
Electronics and Communications Engineering Department
Analysis of Continuous-Time Signals (ELC 2030) – Fall 2022

Project

It is required to use Matlab, or any other programing tool, to perform the following tasks:

1. **Image compression**

   Background:

   The 1D Fourier analysis represents the signal as a weighted sum of sinusoidals or complex exponentials with different frequencies. Similarly, the 2D discrete cosine transform (2D DCT) represents the 2D signal (e.g. an image or a block of an image) as a weighted sum of images with different spatial frequencies, as shown in Fig. 1. Spatial frequencies refer to the rate of variation of pixel values with respect to space coordinates.

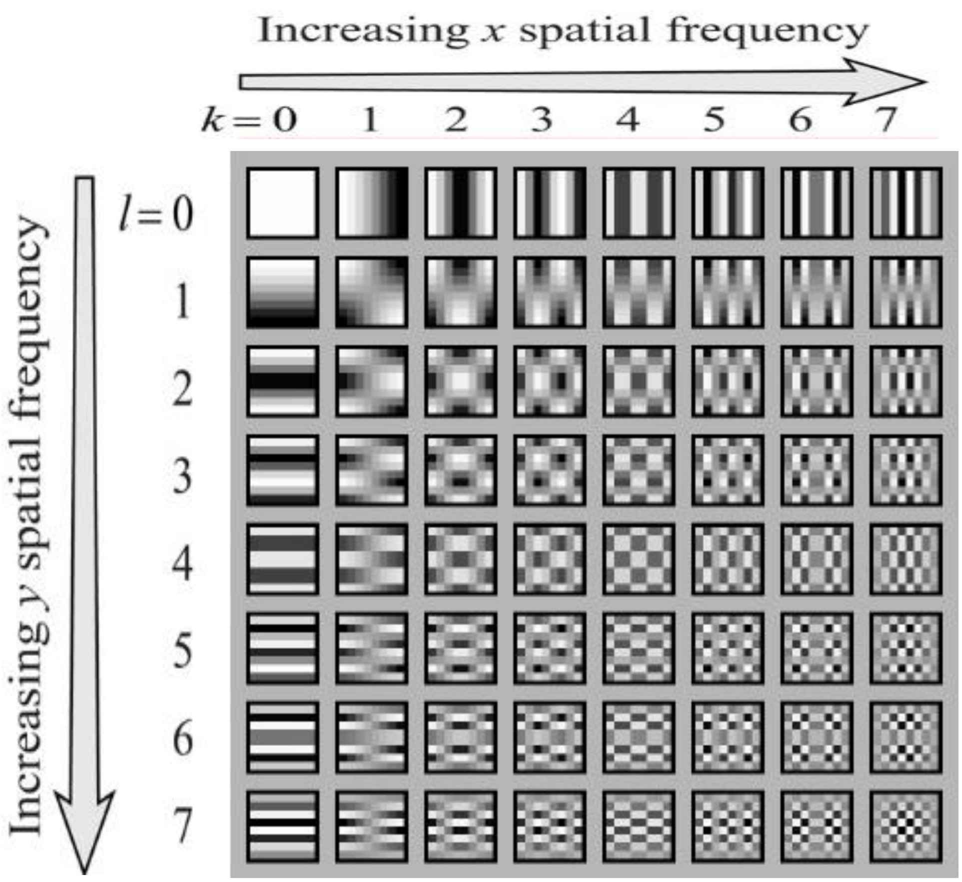

Fig. 1 DCT basis images

The top left corner is DC. Images towards the top left have low horizontal and vertical spatial frequencies. Images towards the right have higher horizontal spatial frequency. Images towards the bottom have higher vertical spatial frequency. The bottom right image has the maximum horizontal and vertical spatial frequencies.

Most of the information of the image is concentrated in the lower spatial frequencies (the top left coefficients have significantly higher values). Additionally, the human eye is more sensitive to lower frequencies and is much less likely to notice the loss of very high spatial frequency components (very fine details). Therefore, ignoring higher spatial frequencies doesn't affect much how the eye perceives the image.

DCT is widely used in image, video, and audio compression, because of its ability to compact most of the energy of a signal into few coefficients.

Task:
In this task, you will perform a simple image compression algorithm. You will read an input image and process each of its color components (red, green, and blue) in blocks of 8×8 pixels. Each block will be converted into frequency domain using 2D DCT and then only few coefficients are retained, while the rest will be ignored.

a) Read the image file 'image1.bmp'. Extract and display each one of its three color components. Repeat the following steps for $m = 1,2,3,4$:
b) To compress the image, process each color component in blocks of 8×8 pixels. Obtain 2D DCT of each block. It will have the same dimensions as the input block, corresponding to the 64 basis images. Retain only the top left square of the 2D DCT coefficients of size $m \times m$. That is, if the DCT coefficients are X[1:8,1:8], retain only the top left $m \times m$ coefficients, $X[1{:}m, 1{:}m]$, assuming that the top left coefficient is $X[1,1]$. The rest of coefficients are ignored.
c) Compare the size of the original and compressed images.
d) Decompress the image by applying inverse 2D DCT to each block. Display the image.
e) The quality of the decompressed image is measured using the Peak Signal-to-Noise Ratio (PSNR), which is defined by

$$PSNR = 10 \log_{10} \frac{peak^2}{MSE}$$

where $peak$ is the peak value for the pixels according to the image datatype (e.g. for uint8 image it is 255). Mean square error (MSE) between the original image and the decompressed image is obtained by subtracting the corresponding pixel values of two images and obtaining the average of the square of all the differences. Obtain the PSNR for each value of m.
f) Plot a curve displaying the PSNR (on the vertical axis) against m (on the horizontal axis). Comment on the resulting graph and quality of images.

2. **Audio spectral analysis and filtering**
Background:
The spectrum of a discrete-time signal (the DTFT) is generally a continuous function of frequency. Therefore, it is not suitable for numerical computations. Instead, we will use the **Discrete Fourier Transform (DFT)**, which is a sampled version of the DTFT.
When a continuous-time signal is sampled at a rate f_s Hz, its spectrum is repeated at multiples of f_s Hz. The N-point DFT of a DT signal $x[n]$ gives N samples of one period of its spectrum. Since we have N samples in a range of frequencies from 0 to f_s Hz, the spacing between the samples is f_s/N. The N-point DFT X[k] is given in the range of $k = 0 \; to \; N-1$ and corresponds to frequencies $f = 0 \; to \; (N-1)f_s/N$. Note that N must be greater than or equal to L, the length of x[n].
We say that $X[k]$ is the DFT of $x[n]$ and $x[n]$ is the inverse DFT (IDFT) of $X[k]$. These two operations are implemented in Matlab using 'fft' and 'ifft' functions, respectively. Fast Fourier transform (FFT) is an efficient algorithm for computing the DFT.

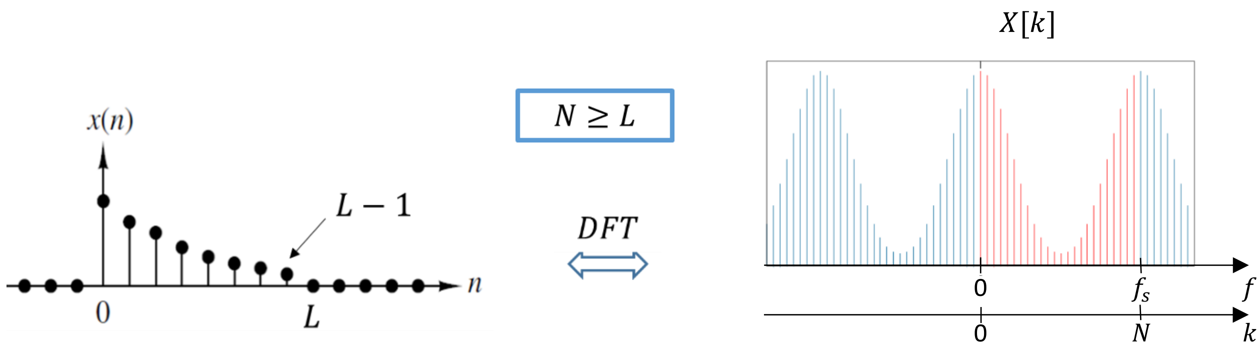

Fig. 2 DFT

Task:
In this task you are required to analyze an audio file which is corrupted by some interference, identify the interference, and filter the audio signal to remove the interference.

a. Use FFT to obtain the magnitude spectrum of the audio file "audio.wav". Plot it against the frequency in Hz. Use the function 'fftshift' to make the zero frequency in the center of the plot. What is the type of the interfering signal?
b. Use a suitable filter of your choice to remove the interfering signal. The filter choice should be based on the spectrum in part (a). (Moving average filter should not be used.)
Give all the filter specifications.
c. Play the filtered audio. (Record it as "filtered.wav")
d. Plot the spectrum of the filtered audio.
e. Plot the frequency response of the filter.
f. Plot the impulse response of the filter. What is the relationship between the impulse response and the frequency response in part (e)?
g. Is the filter causal or not? Explain your answer.

h. Play the filtered audio at twice the speed. Plot the new spectrum. What do you notice on the spectrum compared to the spectrum of the normal speed file?

Hint: You may use the Matlab filter designer tool “filterDesigner” to design the filter and export it as an object. Then use the “filter” function to filter the audio using the designed filter. If used, add a screenshot of the design page to the report. FilterDesigner can also be used to generate the response plots of the filter.

Useful Matlab functions: fft, fftshift, filter, freqz, impz

Deliverables:

1. One **uncompressed pdf** project report containing:
    a. Explanation of your work.
    b. All the required results and answers to questions.
    c. All the required figures. Label your figures properly.
    d. All the codes, included at the end.
2. One zip file containing all the codes, filtered audio file, and compressed images.

Instructions:

- You can work in teams up to 2 members per team.
- Any copied results or codes will result in zero grade for both teams.
- Code in the report should be supplied as text, not as screenshots.

Due date: December 30, 2022, at 11:59 pm.