Bildverarbeitung I (Prof. Schilling)

WS 2023/2024

Assignment 3

Remarks

Please submit your exercises in ILIAS before 23:55 on the closing date. At least one member of the group must be present at our biweekly tutorial, beeing prepared to explain *each* exercise. Random groups will be asked to present their solutions. Stick to the submission procedure described in Assignment 1.
Reminder: If there is a built-in function for an algorithm, you are supposed to implement, do *not* use it. If you are asked to explain your results, hand your answers in as a pdf-document.

Exercise 6: Discrete 1D Fourier Transform [8 points]

The code for all subtasks of this exercises is located in `exercise_06.py`.

a) DFT [2 points]: Complete the function `dft_1d` that transforms a one-dimensional signal to the frequency domain. If your implentation is correct, the resulting plot should look like Figure 1 (especially the Frequency Domain). Do *not* use `np.fft.fft`.

b) Inverse DFT [2 points]: Complete the function `idft_1d` that transforms a one-dimensional signal represented in frequency domain back to the time domain. Do *not* use `np.fft.ifft`.

c) Simple denoising [2 points]: Now you will use `dft_1d` and `idft_1d` to reconstruct a signal that was disturbed by normally distributed random noise. Complete the function `dft_1d_denoise` that filters out all frequencies with amplitudes lower than a certain threshold to denoise the signal.

d) Mean filtering [2 points]: In this task you will implement the same filter in two diffent ways. Complete the functions `box_filter_1d_time` (applying a mean filter in the spatial domain) and `box_filter_1d_freq` (applying a mean filter in the frequency domain). The parameter w defines the size of the boxes, which should range from $x-w$ to $x+w$. Compare the results and explain.

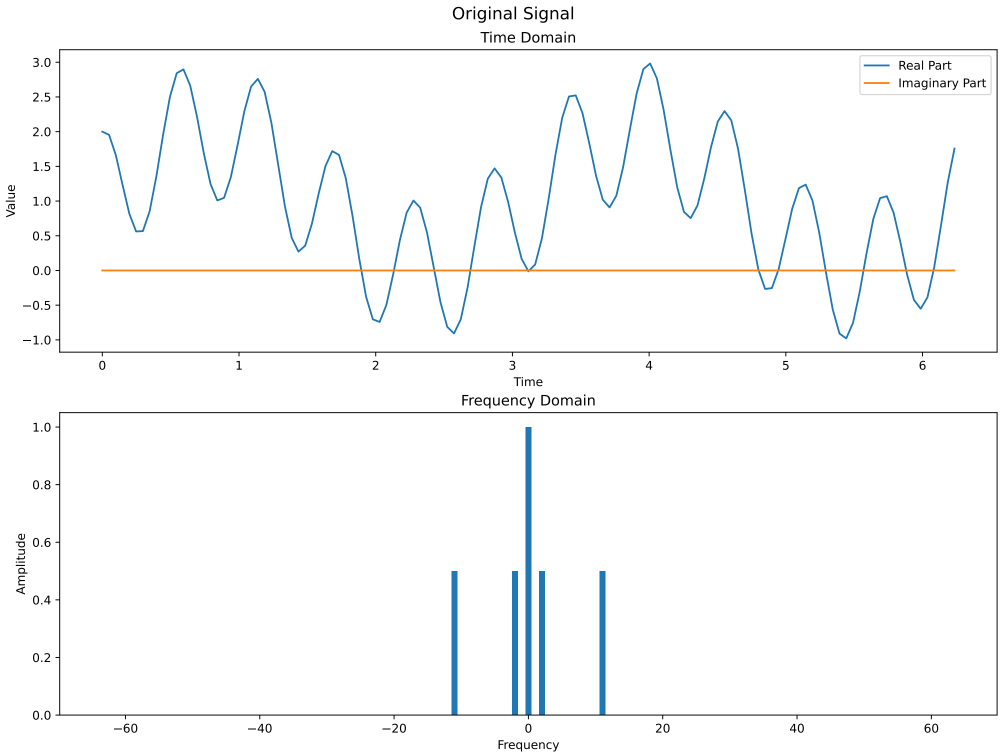

Figure 1: The upper plot shows two superimposed sine waves with different frequencies. The frequencies present in the signal are centered around zero and plotted below. The bars' heights correspond to the amplitudes. Note that this plot does not contain any information about the phase shifts.

Exercise 7: Discrete 2D-FFT [4 points]

In this exercise (`exercise_07.py`), the two dimensional signal shown in Figure 2 should be filtered in the frequency domain. Use the `numpy` functions `np.fft.fft2`, `np.fft.ifft2`, `np.fft.fftshift` and `np.fft.ifftshift` to complete the following tasks.

a) Ideal low pass filter [2 points]: Complete the function `ideal_low_pass_filter` that implements an ideal low-pass filter in the frequency domain.

b) Ideal high pass filter [2 points]: Complete the function `ideal_high_pass_filter` that implements an ideal high-pass filter in the frequency domain.

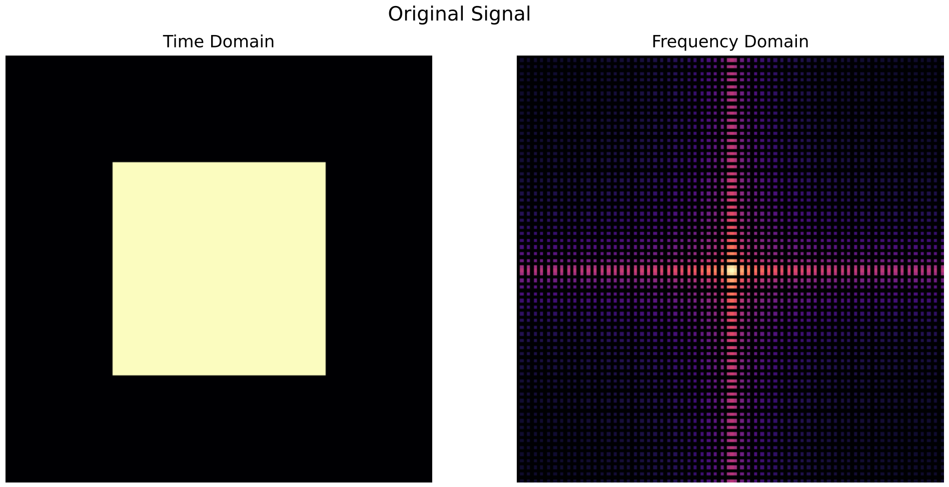

Figure 2: The left plot shows the signal in time domain. The right side shows the signals frequency spectrum. Note that the amplitudes are transfered to log-scale for a better visualization.