

Assignment 5: CS 663

Due: 13th November before 11:55 pm

Remember the honor code while submitting this (and every other) assignment. You may discuss broad ideas with other students or ask me for any difficulties, but the code you implement and the answers you write must be your own. We will adopt a zero-tolerance policy against any violation.

Submission instructions: Follow the instructions for the submission format and the naming convention of your files from the submission guidelines file in the homework folder. Please see `assignment5_DFT.rar`. Upload the file on moodle before 11:55 pm on 13th November. Only one student per group needs to upload the assignment. We will not penalize submission of the files till 1 pm on 16th November. No late assignments will be accepted after this time. Please preserve a copy of all your work until the end of the semester.

1. Suppose you are standing in a well-illuminated room with a large window, and you take a picture of the scene outside. The window undesirably acts as a semi-reflecting surface, and hence the picture will contain a reflection of the scene inside the room, besides the scene outside. While solutions exist for separating the two components from a single picture, here you will look at a simpler-to-solve version of this problem where you would take two pictures. The first picture g_1 is taken by adjusting your camera lens so that the scene outside (f_1) is in focus (we will assume that the scene outside has negligible depth variation when compared to the distance from the camera, and so it makes sense to say that the entire scene outside is in focus), and the reflection off the window surface (f_2) will now be defocussed or blurred. This can be written as $g_1 = f_1 + h_2 * f_2$ where h_2 stands for the blur kernel that acted on f_2 . The second picture g_2 is taken by focusing the camera onto the surface of the window, with the scene outside being defocussed. This can be written as $g_2 = h_1 * f_1 + f_2$ where h_1 is the blur kernel acting on f_1 . Given g_1 and g_2 , and assuming h_1 and h_2 are known, your task is to derive a formula to determine f_1 and f_2 . Note that we are making the simplifying assumption that there was no relative motion between the camera and the scene outside while the two pictures were being acquired, and that there were no changes whatsoever to the scene outside or inside. Even with all these assumptions, you will notice something inherently problematic about the formula you will derive. What is it? [8+7 = 15 points]
2. Consider a 1D image (for example, a single row from a 2D image). You know that given such an image, computing its gradients is trivial. An inquisitive student frames this as a convolution problem to yield $g = h * f$ where g is the gradient image (in 1D), h is the convolution kernel to represent the gradient operation, and f is the original 1D image. The student tries to develop a method to determine f given g and h . What are the fundamental difficulties he/she will face in this task? Justify your answer. You may assume appropriate boundary conditions. Now consider that you are given the gradients of a 2D image in the X and Y directions, and you wish to determine the original image. What are the difficulties you will face in this task? Justify your answer. Again, you may assume appropriate boundary conditions. [10+10 = 20 points]
3. Consider the image with the low frequency noise pattern shared in the homework folder in the form of a .mat file. Your task is to (a) write MATLAB code to display the log magnitude of its Fourier transform, (b) to determine the frequency of the noise pattern by observing the log magnitude of the Fourier transform and guessing the interfering frequencies, and (c) to design and implement (in MATLAB) an ideal notch filter to remove the interference(s) and display the restored image. To this end, you may use the `fft2`, `ifft2`, `fftshift` and `ifftshift` routines in MATLAB. [15 points]
4. Consider the `barbara256.png` image from the homework folder. Implement the following in MATLAB: (a) an ideal low pass filter with cutoff frequency $D \in \{40, 80\}$, (b) a Gaussian low pass filter with $\sigma \in \{40, 80\}$. Show the effect of these on the image, and display all images in your report. Display the frequency response (in log

Fourier format) of all filters in your report as well. Comment on the differences in the outputs. Make sure you perform appropriate zero-padding! [20 points]

5. Read Section 1 of the paper ‘An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration’ published in the IEEE Transactions on Image Processing in August 1996. A copy of this paper is available in the homework folder. Implement the technique in Equation 3 of the paper to align two images which are related to each other by a 2D in-plane translation. Test your implementation on images I and J as follows. I is a 300×300 image containing a 50×70 white rectangle (intensity 255) whose top-left corner lies at pixel $(50, 50)$. All other pixels of I have intensity 0. The image J is obtained from a translation of I by values $(t_x = -30, t_y = 70)$. Verify carefully that the predicted translation agrees with the ground truth translation values. Repeat the exercise if I and J were treated with iid Gaussian noise with mean 0 and standard deviation 20. In both cases, display the logarithm of the Fourier magnitude of the cross-power spectrum in Equation 3 of the paper. What is the time complexity of this procedure to predict translation if the images were of size $N \times N$? How does it compare with the time complexity of pixel-wise image comparison procedure for predicting the translation?

Also, briefly explain the approach for correcting for rotation between two images, as proposed in this paper in Section II. Write down an equation or two to illustrate your point. [8+7=15 points]

6. Consider two different Laplacian filter kernels $k_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ and $k_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$. Write down the formulae for their N, N -point Discrete Fourier Transforms in the report. Compute their N, N -point Discrete Fourier Transforms in MATLAB with $N = 201$, i.e. where the spatial and frequency indices range from -100 to 100 in both canonical directions. Display the magnitude of the DFT on a log scale using the `imshow` and `surf` functions in MATLAB along with a colorbar. Besides the plots, include the code snippet for the DFT computation and display in the report. Comment on the difference in the Fourier transforms of the two kernels. [10+5=15 points]