HW#1

Due 23:59 on 20th of November, Sunday

Q1) In this problem, you will explore various ways of computing 2D convolution between an object $x(n_1, n_2)$ and an impulse response $h(n_1, n_2)$ (i.e. point spread function (PSF)). Together with the results and answers, **all Matlab codes should also be provided** to get credit.

(a) Figure 1 shows a simple object $x(n_1, n_2)$ and its image after convolution with the PSF $h(n_1, n_2)$. The nonzero entries of both matrices (as shown in nonblack pixels) are:

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \textit{and} \quad \begin{bmatrix} & \frac{1}{4} & \frac{1}{4} & \\ \frac{1}{4} & 1 & 1 & \frac{1}{4} \\ \frac{1}{4} & 1 & 1 & \frac{1}{4} \\ & \frac{1}{4} & \frac{1}{4} & \end{bmatrix}$$

respectively. Can you infer what the PSF is? Show your work. Also verify it in MATLAB with

abs(ifft2(fft2(*x*).*fft2(*h*)))

where each matrix contains corresponding 8*x*8 pixel values.

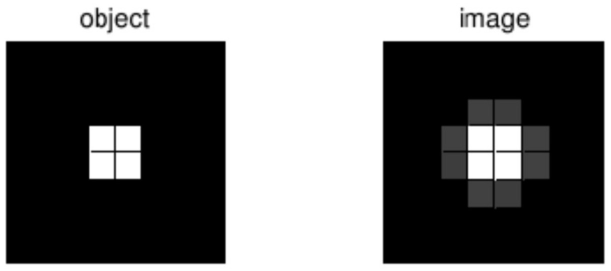

Figure 1: Object and image in 8x8 pixels

(b) Is the PSF $h(n_1, n_2)$ separable? Why or why not?

(c) Find analytically the 2D DTFT of this PSF, i.e. the frequency response of the underlying LSI system. Plot the magnitude of this DTFT and comment on the effects of this PSF on the input object.

(d) Compute and plot the 2D DFT of this PSF analytically. Then verify it in Matlab using fft2. Compare these results with part (c), and discuss the relations between them.

(e) The two-dimensional convolution between an object $x(n_1, n_2)$ and a PSF $h(n_1, n_2)$, i.e. $y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2)$, can be written in matrix form as $y = Cx$ where y and x are vectorized versions of $y(n_1, n_2)$ and $x(n_1, n_2)$, respectively, i.e.

$$[x(n_1,n_2)] = \begin{bmatrix} x(1,1) & x(1,2) & \cdots & x(1,L_2) \\ x(2,1) & x(2,2) & \cdots & x(2,L_2) \\ \vdots & \vdots & \vdots & \vdots \\ x(L_1,1) & x(L_1,2) & \cdots & x(L_1,L_2) \end{bmatrix} \Rightarrow x = \begin{bmatrix} x(1,1) \\ \vdots \\ x(L_1,1) \\ ---- \\ x(1,2) \\ \vdots \\ x(L_1,2) \\ ---- \\ \vdots \end{bmatrix}$$

In MATLAB this operation is performed using the colon operator \:". The opposite operation taking a vector back to an image is done using the MATLAB function *reshape*. When the vectors y and x are defined this way for the images $y(n_1,n_2)$ and $x(n_1,n_2)$, the entries of the matrix C which captures this convolution as $y = Cx$ are as follows:

$$C = \begin{bmatrix} s_0 & 0 & \cdots & 0 \\ s_1 & s_0 & & \vdots \\ \vdots & \vdots & & \\ s_{P_2-1} & s_{P_2-2} & & \\ 0 & s_{P_2-1} & \ddots & s_0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & s_{P_2-1} \end{bmatrix}$$

The matrix is ”doubly block Toeplitz” as each block entry itself is also a Toeplitz matrix given by

$$s_i = \begin{bmatrix} h(0,i) & 0 & \cdots & 0 \\ h(1,i) & h(0,i) & & \vdots \\ \vdots & \vdots & & \\ h(P_1-1,i) & h(P_1-2,i) & & \\ 0 & h(P_1-1,i) & \ddots & h(0,i) \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & h(P_1-1,i) \end{bmatrix}$$

whose entries are from the *i*th column of the two-dimensional PSF. Note that the linear convolution matrix C is an $(L_1 + P_1 - 1)(L_2 + P_2 - 1) \times L_1 L_2$ matrix when $x(n_1,n_2)$ is $L_1 \times L_2$ image and $h(n_1,n_2)$ is $P_1 \times P_2$. Construct the matrix with the PSF you find in part (a), and verify that you get the same image data as before by matrix multiplication.

(f) Multiplication by a Toeplitz matrix s_i corresponds to 1-D linear convolution and hence can be efficiently computed with a one-dimensional FFT. In MATLAB notation:

s_i*f = ifft(fft(h(:,i)).*fft(f))

Implement the two-dimensional convolution with one-dimensional FFTs, and verify that you get the same image data as before.

(g) Download "mandrill.bmp" from the web and convert it to a grayscale image. Use your one-dimensional FFT implementation to get its convolution with the PSF in part (a). You should get a similar picture as shown in figure 2. You may have noticed that it appears brighter than the original image which is physically nonrealistic. Normalize the PSF so that it does not amplify the energy of an input signal. Redo the simulation and comment on the difference between the original data and the convolved one.

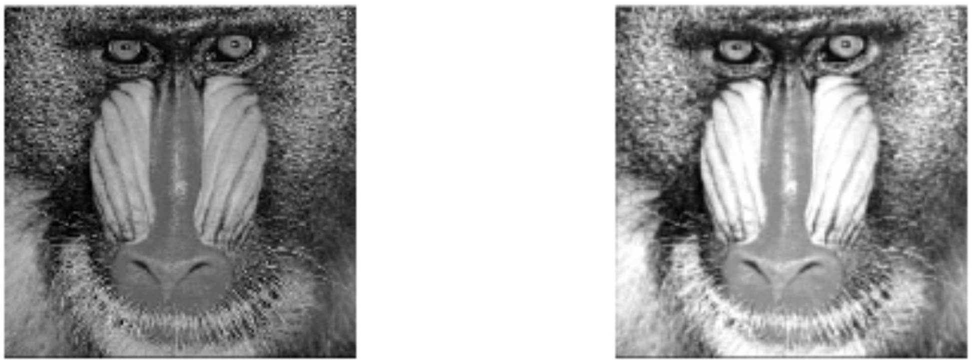

Figure 2: Image of a mandrill and its convolution with a PSF

Q2) Suppose an analog signal $s(t_1, t_2)$ has Fourier transform $S(f_1, f_2)$ whose support is circular, that is $S(f_1, f_2) = 0$ for ${f_1}^2 + {f_2}^2 \geq 1$.

(a) Based on Nyquist sampling theorem, state the range of sampling intervals for sampling this signal on a rectangular grid to enable exact recovery from the samples.
(b) If we are allowed to sample the signal on an arbitrary grid, find the sampling grid that will enable exact recovery from the minimum possible number of samples.

Q3) In this question, your task is to obtain some of the results illustrated in the lectures. In each part, you are referred to one or more slides used in the lectures and asked to generate the results shown in the slides in Matlab (or Phyton). Slide numbers given are the exact page numbers in the documents (not the written slide numbers).

In your solution, you should briefly explain how your implementation works and provide figures that show your results. **All Matlab codes should also be provided** to get credit. You should not use any built-in Matlab/Phyton function to directly perform the required task unless stated. General-purpose functions like fft2, ifft2, conv2, imagesc, image, etc. can be used if needed.

(a) Plot the centered and non-centered DFT – slide 78 & 79
Image to be used: 256x256 grayscale Lena image (shared in ODTUClass)

(b) Upsample 128x128 image to 256x256 image – slide 98-101
Image to be used: 256x256 grayscale Lena image - crop a 128x128 square out of Lena image (that contains the face)
Compare the upsampled image with the original 256x256 Lena image