E9 241 DIP - Assignment 03

October 19, 2021

Due Date: October 25, 2021 **Total Marks:** 80 + 20

Instructions:

- For all the questions, write your own functions. Use library functions for comparison only.
- Your function should take the specified parameters as inputs and output the specified results.
- Also provide the wrapper/demo code to run your functions. Your code should be self contained i.e. one should be able to run your code as-is without any modifications.
- Vectorize your code. Non-optimized code may be penalized.
- For python, if you use any libraries other than numpy, scipy, scikit-image, opencv, pillow, matplotlib, pandas and default modules, please specify the library that needs to be installed to run your code.
- Along with your code, also submit a PDF with all the results and answers to subjective questions, if any.
- Put all your files into a single zip file and submit the zip file. Name the zip file with your name.

Q1. **Frequency Domain Filtering:**

(a) Generate a $M \times N$ sinusoidal image $\sin(2\pi u_0 m/M + 2\pi v_0 n/N)$ for $M = N = 1001, u_0 = 100, v_0 = 200$ and compute its DFT. To visualize the DFT of an image take logarithm of the magnitude spectrum.
Note: Fast Fourier Transform (FFT) is an algorithm which is used for efficient computation of DFT of discrete signals. You can use matlab (or python) built in function for computing the FFT.

(b) Filter the image `characters.tif` in the frequency domain using an ideal low pass filter (ILPF). The expression for the ILPF is

$$H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) > D_0 \end{cases} \tag{1}$$

where D_0 is a positive constant referred to as the cut-off frequency and $D(u,v)$ is the distance between a point (u,v) in the frequency domain and the center of the frequency rectangle, i.e., $D(u,v) = \sqrt{(u-P/2)^2 + (v-Q/2)^2}$, where P and Q are the number of rows and columns in the image. What artefacts do you notice in the image obtained by computing the inverse DFT of the filtered image?
Note: The filter given above is centred in frequency domain. To use such centred filters, you will either need to shift the filter to $(0,0)$ (by using fftshift in matlab or corresponding function in python) or center the DFT of the image. To center the DFT of the image, you can either shift the DFT of the image, or scale each image pixel $I(x,y)$ by -1^{x+y} before computing its DFT. If you center the DFT of the image, then you will need to compensate for it by multiplying the image obtained from inverse DFT of the filtered image by -1^{x+y}.

(c) Filter the image characters.tif in the frequency domain using the Gaussian low pass filter given by

$$H(u, v) = \exp(-D^2(u, v)/2D_0^2) \tag{2}$$

where all the terms are as explained in part b. For $D_0 = 100$, compare the result with that of the ILPF.

(10+10+10=**30M**)

Q2. Image Deblurring: Deblur the images `Blurred_LowNoise.png` (White Noise Standard Deviation (σ)= 1) and `Blurred_HighNoise.png` ($\sigma = 10$) which have been blurred by the kernel `BlurKernel.mat` using

(a) Inverse filtering: Simple inverse filtering may lead to amplification of noise (why?). To mitigate amplification of high frequency noise, set the inverse filter fft values to 0 wherever blur filter fft values are below a threshold t. Set $t = 0.1$.
<u>Note</u>: The blur kernel given is centered at $(10, 25)$ in matlab convention and $(9, 24)$ in python convention. In other words, the maximum value 0.02 corresponds to $(0, 0)$ of the filter. So, before computing DFT, you need to shift the filter such that $(10, 25)$ moves to $(1, 1)$, $(10, 24)$ moves to $(1, N)$, $(9, 25)$ moves to $(M, 1)$ and $(9, 24)$ moves to (M, N) and so on in matlab convention. In python convention $(9, 24)$ moves to $(0, 0)$, $(9, 23)$ moves to $(0, N-1)$, $(8, 24)$ moves to $(M-1, 0)$ and $(8, 23)$ moves to $(M-1, N-1)$ and so on. M, N are the height and width of the blurred image respectively.

(b) Wiener filter: You can assume the PSD of the white noise is equal to σ specified. For signal PSD, use power law

$$S_f(u, v) = \frac{k}{\sqrt{u^2 + v^2}}. \tag{3}$$

Use $k = 10^5$.
<u>Note</u>: Recall that $S_f(u, v)$ is periodic and $S_f(-u, -v) = S_f(M-u, N-v)$ where M, N are height and width of the blurred image respectively .

<u>Note</u>: `BlurKernel.mat` is a matlab workspace variable file. You can load this file using 'load(BlurKernel.mat)' in matlab provided that the file is stored in your working directory. For python, see `https://stackoverflow.com/a/874488/3337089`. (10+10=**20M**)

Q3. Image Denoising:

(a) Denoise the image `noisy_book1.png` corrupted by the impulsive noise by a mean and Median filter with same spatial neighborhood size and compare the results.

(b) Use the bilateral filter to denoise the image `noisy_book2.png` which has been corrupted by the Gaussian noise with zero mean and standard deviation of 5. Compare the results with the Gaussian smoothing.

(10+20=**30M**)

Q4. Bonus Question:

(a) Consider the 5×5 discrete Laplacian filter given by:

$$g = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Suppose we design the Laplacian in frequency domain as $H(u, v) = K * (u^2 + v^2)$ for $u, v = \{-2, -1, 0, 1, 2\}$. Let $h(m, n)$ be the IDFT of $H(u, v)$. Find K that minimizes

$$\sum_m \sum_n (h(m, n) - g(m, n))^2$$

.

(b) Consider another 5×5 discrete Laplacian filter given by:

$$g = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & -24 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Compute the value of K in this case. Comment on the value of K that you obtained in (a) and (b).

(20M)