E9 241: DIGITAL IMAGE PROCESSING

ASSIGNMENT 3; DUE SEPTEMBER 5/6, 2019

TOPIC: THE 2-D DISCRETE FOURIER TRANSFORM (2-D DFT)

HANDWRITTEN TASKS (DUE: SEPTEMBER 5, 5.15 PM)

(1) Show that $\left\{e^{-\mathrm{j}m\omega_1 x-\mathrm{j}n\omega_2 y}, x, y \in \mathbb{R}, m, n \in \mathbb{Z}, \omega_1, \omega_2 \in [-\pi, +\pi]\right\}$ constitutes an orthogonal set over the region: $x \in \left[-\dfrac{\pi}{\omega_1}, +\dfrac{\pi}{\omega_1}\right], y \in \left[-\dfrac{\pi}{\omega_2}, +\dfrac{\pi}{\omega_2}\right]$.

(2) Show that $\left\{e^{-\mathrm{j}\frac{2\pi k n}{N}}, k, n = 0, 1, 2, \ldots, N-1, N \in \mathbb{N}\right\}$ constitutes an orthogonal set over $n \in [0, N-1]$. Consider an $N \times N$ matrix F whose (k, n)-element is $e^{-\mathrm{j}\frac{2\pi k n}{N}}, k, n = 0, 1, 2, \ldots, N-1$. Determine the adjoint of F.

(3) Consider two $N \times N$ images $f[k, \ell]$ and $g[k, \ell], k, \ell = 0, 1, 2, \ldots, N-1$ whose 2-D DFTs of size $N \times N$ are denoted by $\widehat{f}[k', \ell']$ and $\widehat{g}[k', \ell'], k', \ell' = 0, 1, 2, \ldots, N-1$, respectively. Show that

$$\sum_{k=0}^{N-1}\sum_{\ell=0}^{N-1} f[k, \ell] g^*[k, \ell] = \frac{1}{N^2}\sum_{k'=0}^{N-1}\sum_{\ell'=0}^{N-1} \widehat{f}[k', \ell']\widehat{g}^*[k', \ell'].$$

The left-hand side is the inner-product between f and g. Similarly, the double summation on the right-hand side is the inner-product between $\widehat{f}$ and $\widehat{g}$. This property shows that the 2-D DFT preserves inner-products between sequences up to a scale factor. Use this result to show that the 2-D DFT is an energy-preserving transformation (*the Parseval property*).

(4) Consider the N-point sequence $\{f[n], n = 0, 1, 2, \ldots, N-1\}$ and its N-point DFT $\{\widehat{f}[k], k = 0, 1, 2, \ldots, N-1\}$. Express the N-point DFT of $\{\widehat{f}[k], k = 0, 1, 2, \ldots, N-1\}$ in terms of $f[n]$. Considering N-length vector representations of the sequences $f[n]$ and $\widehat{f}[k]$, express the result using matrix/vector notation.

PROGRAMMING TASKS (DUE SEPTEMBER 6)

Remember: DFT and IDFT are the **transforms**. FFT and IFFT are **fast algorithms** to implement the DFT and IDFT, respectively, and not transforms themselves.

(1) This exercise would help you understand how to implement the 2-D DFT efficiently using the 1-D fast Fourier transform (FFT) algorithm. Effectively, you would develop a **2-D FFT**.
Write a Matlab/Python script that accepts an $M \times M$ image and a number N as inputs and implements a 2-D DFT of size $N \times N$. Both M and N are powers of 2. There is no order relation between M and N, i.e., M could be greater than N or vice versa. Your program must allow for M and N to be different. You are allowed to use the built-in 1-D fast Fourier transform (FFT) implementation. The output should be the $N \times N$ 2-D DFT. Report results for the two cases: (i) $N = 2M$; and (ii) $N = M/2$. Use the *Zone plate* image to generate results. Display the *Zone plate* image (title: ZONE PLATE IMAGE), the magnitude of the 2-D DFT (title: 2-D DFT MAGNITUDE), and the phase of the 2-D DFT (title: 2-D DFT PHASE).
Titbit: A zone plate is a device that is used to focus electromagnetic waves. However, unlike lenses or curved mirrors, which use refraction or reflection, respectively, a zone plate uses diffraction for focusing an incident wave.

(2) This exercise tests your ability to exploit the properties of the DFT to implement the **2-D inverse FFT**.
Use the 2-D FFT module developed in (1) as a black-box to implement the 2-D inverse FFT (IFFT). You could add processing blocks before or after the 2-D FFT block, but you should not change the contents of the 2-D FFT block. Your program should accept the 2-D DFT of an image as the input and output the image in the spatial domain. The input size is $N \times N$ and the output size is $M \times M$. There is no order relation between M and N, i.e., M could be larger than N or vice versa. Your program should allow for M and N to be different. Ideally, if you cascade the 2-D FFT and 2-D IFFT blocks, both of size $M \times M$, you should get back the image that went into the 2-D FFT block. Use the 2-D DFT of the *Luna* image to report your results. Display the *Luna* image (title: LUNA IMAGE), the output of the 2-D IFFT block, which is the reconstructed *Luna* image (title: RECONSTRUCTED LUNA IMAGE), and the difference

between the original *Luna* image and the reconstructed one (title: ERROR IMAGE). What is the source of the error?

Titbit: *Luna* (as opposed to *Lenna* :)) is a part of an image of the lunar surface captured by the Terrain Mapping Camera (TMC) of Chandrayaan 2 and was shared by Indian Space Research Organization (ISRO) on August 23, 2019. The name *Luna* was coined by Mr. Ashutosh Gupta (ISRO-SAC, Ahmedabad), an alumnus of Spectrum Lab, EE, IISc.

(3) This exercise would help you understand the **effect of the DFT size** in performing the forward and inverse transformations.

Using the modules developed in Exercises 1 and 2, write a Matlab/Python script to do the following in order: $M \times M$ image $\rightarrow$ $N \times N$ DFT $\rightarrow$ $P \times P$ IDFT. M and N are powers of 2. Use the *Shepp-Logan phantom* to report your results. Report the results for the following cases: (i) $P = N = M$, (ii) $P = N = 2M$, (iii) $2P = N = M$, (iv) $P = N = M/2$. These four cases must appear as the options for a radio button. Display the input image (title: SHEPP-LOGAN PHANTOM) and the output image (title: RECONSTRUCTED SHEPP-LOGAN IMAGE). Based on the selected option, the corresponding results must be computed dynamically and displayed. Explain why the output appears the way it does in each case.

Titbit: Named after its inventors, the Shepp-Logan phantom (1974) has now become a standard test image that serves as a model of a cross-section of the human head in the development and testing of image reconstruction algorithms.

(4) This **magnitude-phase swapping** experiment would help you understand the relative importance of the magnitude vs. phase spectra of images.

Consider two $M \times M$ images f_1 and f_2. Compute their $M \times M$ point DFTs. Next, take the DFT magnitude of f_1 and combine it with the DFT phase of f_2 and compute the $M \times M$ IDFT – call the result f_3. Similarly, consider the other combination (DFT phase of f_1 and DFT magnitude of f_2) and compute the $M \times M$ IDFT – call the result f_4. Display the results: (i) f_1 (title: IMAGE 1), (ii) f_2 (title: IMAGE 2), (iii) f_3 (title: OUTPUT - SWAP 1), and (iv) f_4 (title: OUTPUT - SWAP 2). Use the *Rajinikanth* and *Amitabh* images as input.

Titbit: Amitabh and Rajinikanth need no introduction, but this is a maiden attempt at using their images in the DIP course assignments.