INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Department of Electronics & Eletrical Communication Engineering

Vision and Intelligent Systems

EC69211 – Image and Video Processing Laboratory

Experiment – 3
Frequency Domain Transformation

Submitted by:

Bbiswabasu Roy (19EC39007)

Jothi Prakash (19EC39023)

CONTENTS

Objective:

1. Implementing Fast Fourier Transform (FFT) and Inverse FFT (IFFT) which can be applied on images with any dimensions
2. Applying FFT on a set of images and getting visualization of magnitude and phase spectrum
3. Read a .tif image, apply given sequence of operations on it and save the modified image into new file

Theory:

Fast Fourier Transform

The Discrete Fourier Transform (DFT) can be written as follows.

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}$$

To determine the DFT of a discrete signal x[n] (where N is the size of its domain), we multiply each of its value by e raised to some function of n. We then sum the results obtained for a given n. If we used a computer to calculate the Discrete Fourier Transform of a signal, it would need to perform N (multiplications) x N (additions) = O(N^2) operations.

To optimize this process in O(NlogN) we use the FFT Algorithm as described below –

Split the data into odd and even terms by using the periodicity property of the Fourier Transform, and then recursively apply the FFT algorithm on the subset of points.

$$\begin{cases} n = 2r & \text{if } even \\ n = 2r+1 & \text{if } odd \end{cases}$$

$$\text{where} \quad r = 1, 2, ..., \frac{N}{2} - 1$$

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}$$

$$x[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r] e^{\frac{-j2\pi k(2r)}{N}} \; + \; x[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] e^{\frac{-j2\pi k(2r+1)}{N}}$$

$$x[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r] e^{\frac{-j2\pi k(2r)}{N}} \; + \; x[k] = e^{\frac{-j2\pi k}{N}} \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] e^{\frac{-j2\pi k(2r)}{N}}$$

$$x[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r] e^{\frac{-j2\pi k(r)}{N/2}} \; + \; x[k] = e^{\frac{-j2\pi k}{N}} \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] e^{\frac{-j2\pi k(r)}{N/2}}$$

$$x[k] = x_{even}[k] \; + \; e^{\frac{-j2\pi k}{N}} x_{odd}[k]$$

Mathematical proof for the Time Complexity

$$\frac{N}{2} \longrightarrow 2\left(\frac{N}{2}\right)^2 + N = \frac{N^2}{2} + N$$

$$\frac{N}{4} \longrightarrow 2\left(2\left(\frac{N}{4}\right)^2 + \frac{N}{2}\right) + N = \frac{N^2}{4} + 2N$$

$$\frac{N}{8} \longrightarrow 2\left(2\left(2\left(\frac{N}{8}\right)^2 + \frac{N}{4}\right) + \frac{N}{2}\right) + N = \frac{N^2}{8} + 3N$$

$$\vdots$$

$$\frac{N}{2^P} \longrightarrow \frac{N^2}{2^P} + pN = \frac{N^2}{N} + (\log_2 N)N = N + (\log_2 N)N$$

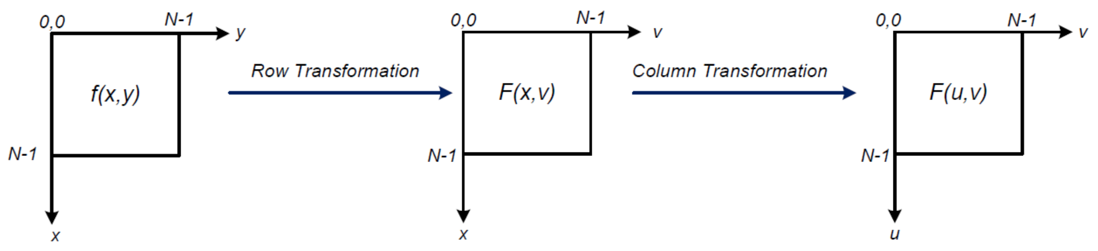
$$\sim O(N + N log_2 N) \sim O(N log_2 N)$$

<u>Fast Fourier Transform – 2D Image</u>

Discrete Fourier Transform in 2D is

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-j\frac{2\pi}{N}(ux+vy)}$$

Since 2D DFT follows the separability property, hence we can apply FFT on the rows first and then apply 2D DFT on the columns to obtain the resultant 2D DFT. Hence, we can apply FFT for computing the respective DFT in 1 Dimension for each axis. The resultant time complexity will be O(N^2logN) which is much better than the brute force of O(N^4)



<u>Inverse Fourier Transform – 2D Image</u>

Inverse Discrete Fourier Transform – 2D Image

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{j\frac{2\pi}{N}(ux+vy)}$$

As we can see the equation nearly identical to the Fourier Transform, only with a change in the sign of the exponent. Similar to the way FFT is computed Inverse FFT is also computed using the same methodology as discussed above.

Algorithm:

Following recursive algorithm was used to compute 1-D FFT of a sequence:

```
1. fft(arr, N, inverse):
2.     arr_even = arr[2*x], arr_odd = arr[2*x+1]
3.     fft(arr_even, N/2)
4.     fft(arr_odd, N/2)
5.     if inverse == false:
6.         w[k] = exp(-j*2πk/N)
7.     else:
8.         w[k] = exp(j*2πk/N)
9.     update arr[k] = arr_even[k] + w[k]*arr_odd[k]
```

Following algorithm is used to visualize the magnitude and phase spectrum of the image:

1. Take the input image name from the user and then read the image using Open CV
2. Covert the Open CV Mat object to a complex pixel_array to perform complex operations
3. Compute the 2D DFT of the pixel_array by first applying FFT along columns and then along rows to get the Fourier Transform coefficients
4. Store the magnitude and phase of the Fourier Transform coefficients in a Open CV Mat object.
5. Store the Mat object in jpg format to view the resultant plots.

Following algorithm is used to perform the desired operations on the .tif image

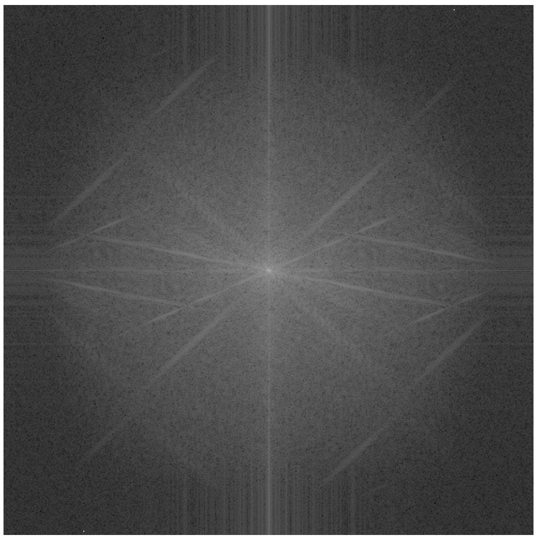
1. Take the input image name from the user and the read the image using Open CV and convert it to a complex pixel_array
2. Multiply the pixel_array with (-1)x+y with x as the row number and y is the column number
3. Compute the 2D DFT using FFT and take the complex conjugate of the resultant transformation
4. Then perform Inverse 2D DFT using FFT of the resultant matrix from previous step.
5. Multiply the real part of the complex pixel_array with (-1)x+y
6. Store the pixel_array in a Mat object and then write in jpg format to view the resultant plots.

<u>Results</u>:

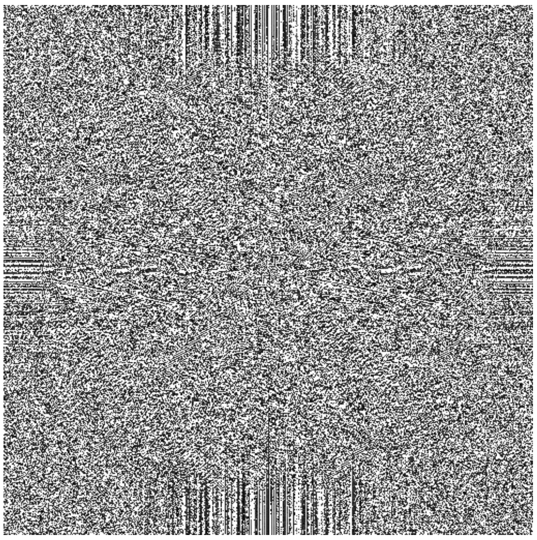
Each of the given images were passed through the FFT function to get the transformed matrix in frequency domain. Then FFT shift was applied to get the low frequency components at the center of the image while the high frequency components at the edges and corner of the image.

However, the magnitude of FFT values took very high values making it difficult to visualize the spectrum. Hence, the log magnitude spectrum with proper scaling was plotted so as to get better contrast of intensity at different parts of the image. The phase spectrum was also plotted after converting it to appropriate scale.

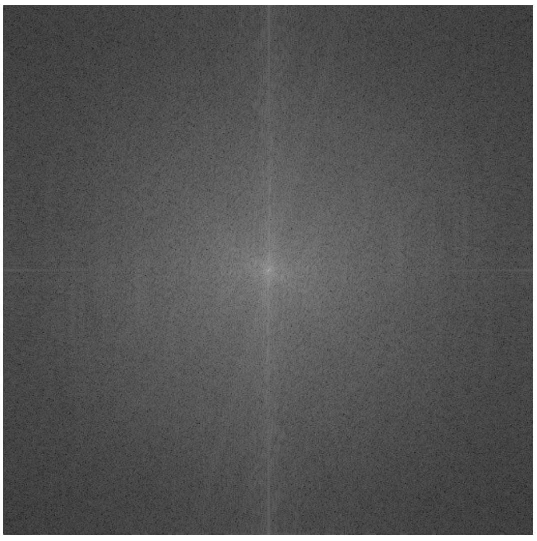
For most images, energy was concentrated in a small region of low frequency. However, for *cameraman.jpg*, we found considerable intensity of magnitude spectrum even at some higher frequency. For *mandril_gray.jpg*, the intensity distribution was significant and almost uniform upto some medium (not very high) frequency and then suddenly it reduced to very low value.



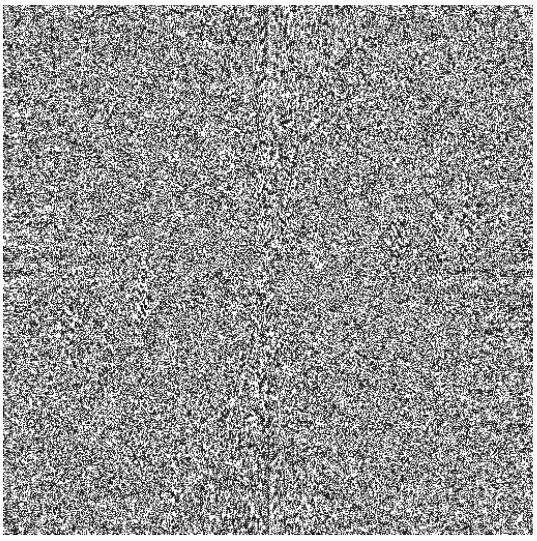
cameraman log magnitude spectrum



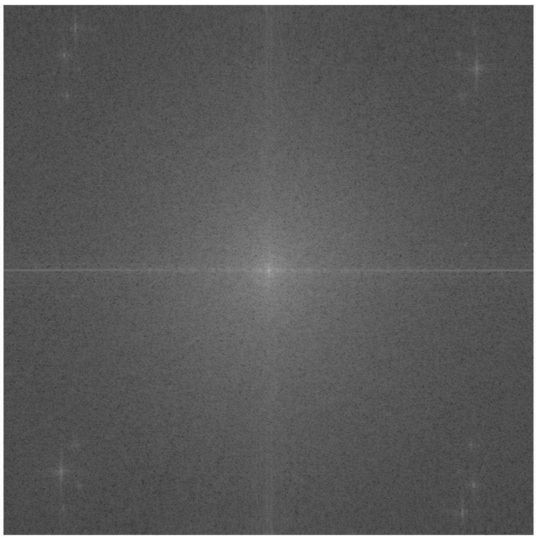
cameraman phase spectrum

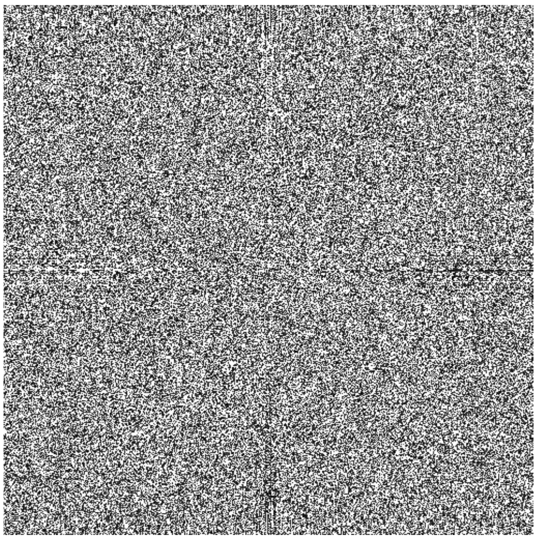


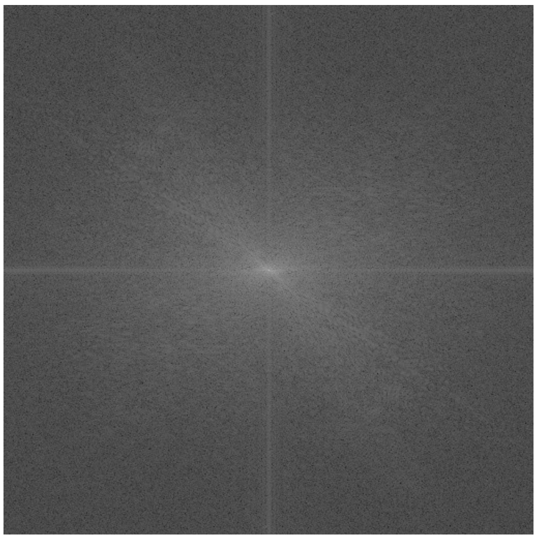
jetplane log magnitude spectrum

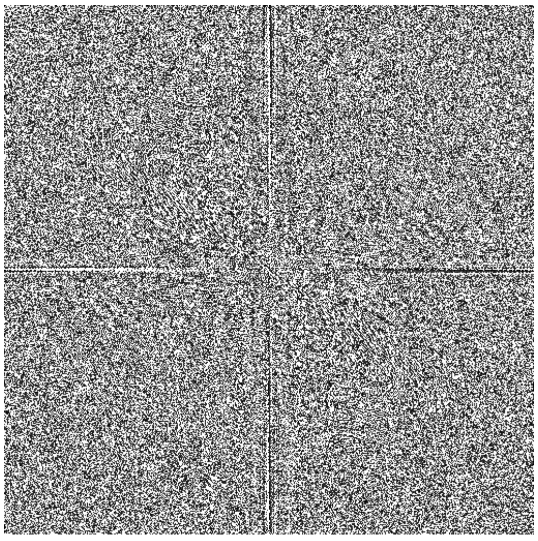


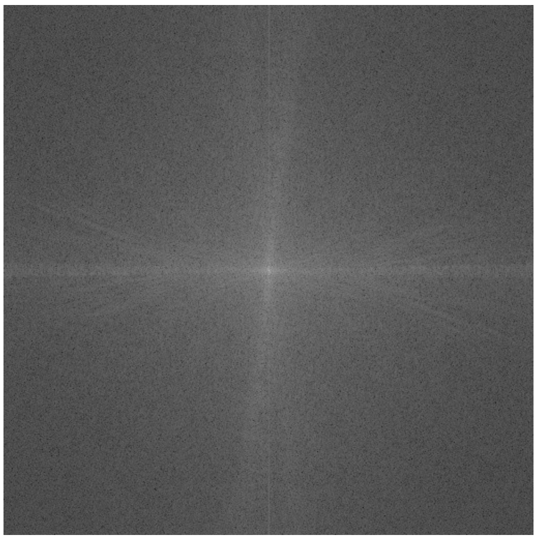
jetplane phase spectrum

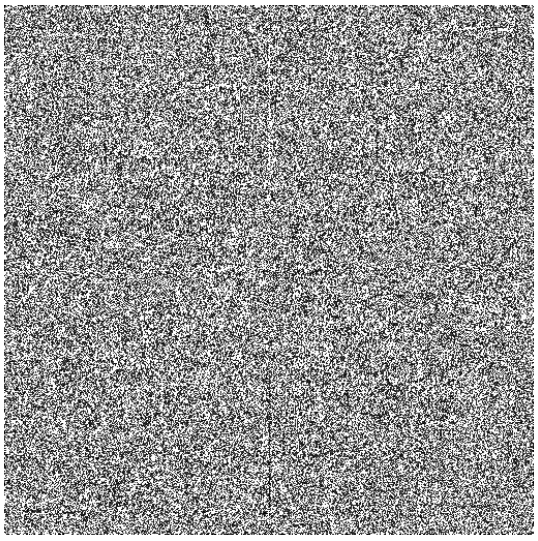

lake log magnitude spectrum

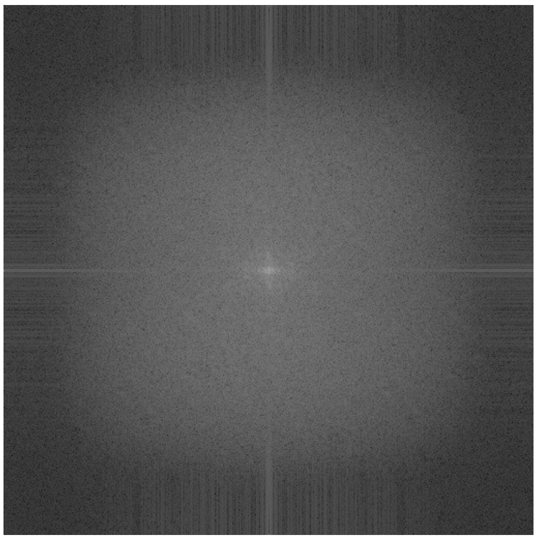

lake phase spectrum

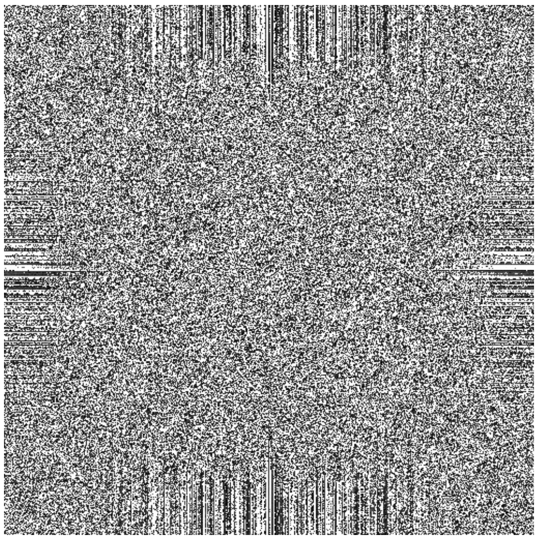

lena_gray_512 log magnitude spectrum

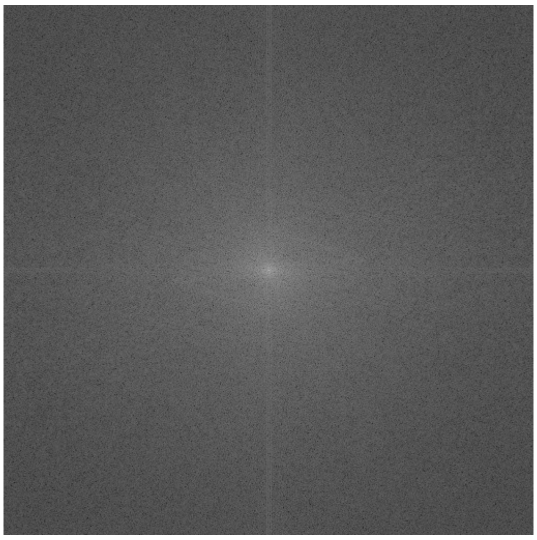

lena_gray_512 phase spectrum

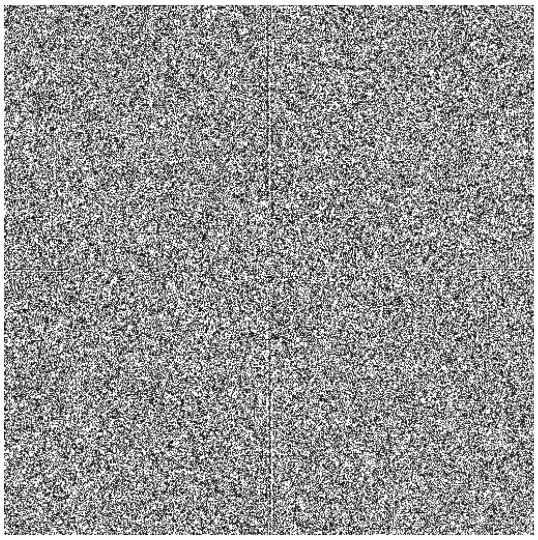

livingroom log magnitude spectrum

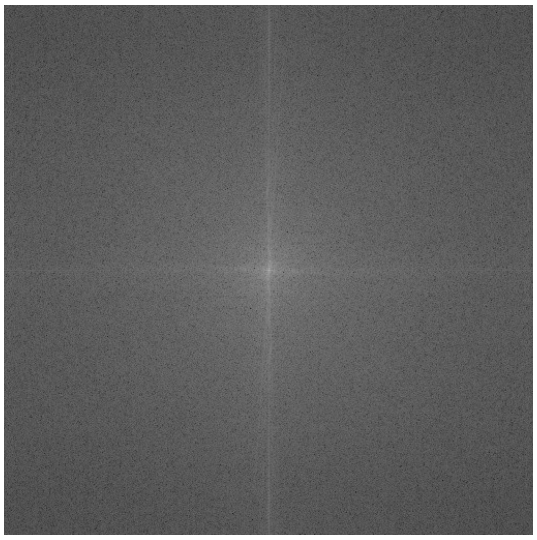

livingroom phase spectrum

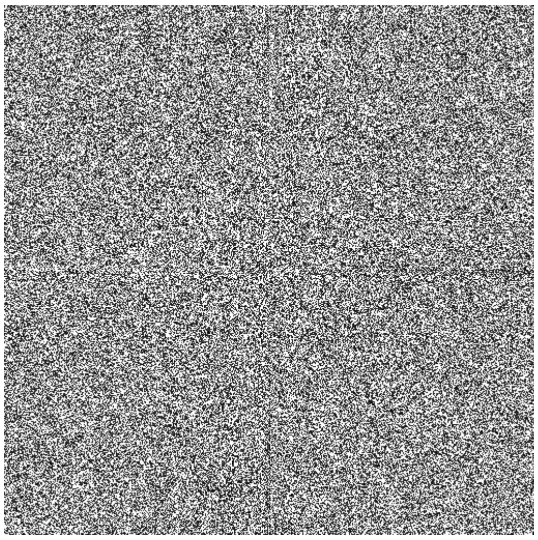

mandril_gray log magnitude spectrum

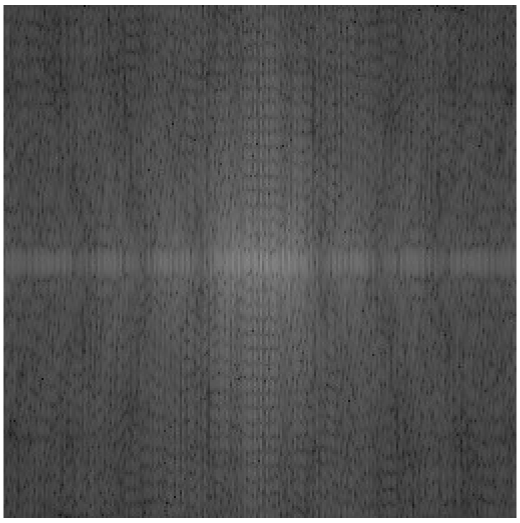

mandril_gray phase spectrum

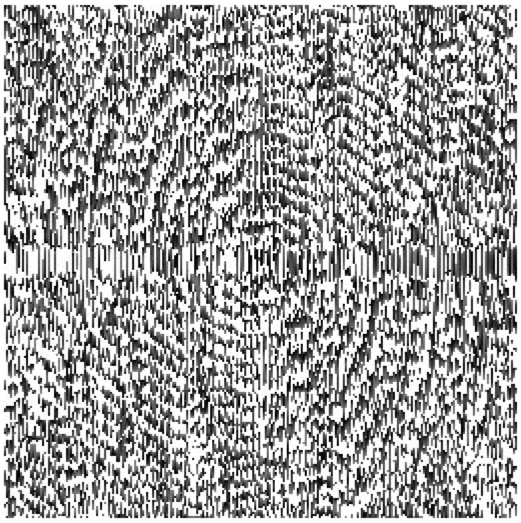

pirate log magnitude spectrum


pirate phase spectrum

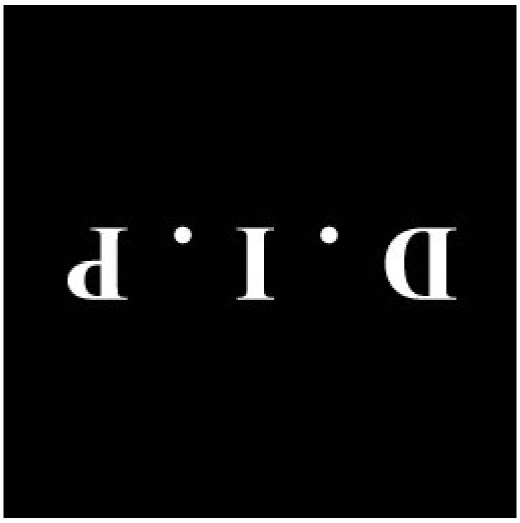

walkbridge log magnitude spectrum


walkbridge phase spectrum


dip.tif log magnitude spectrum


dip.tif phase spectrum

When the sequence of operation as mentioned in Q2 were applied, it was found that the pixels (and hence the text "D . I . P") flipped about the center of the image.



dip.tif after applying given sequence of operations

If we consider the above operations for a 1-D signal, we obtain the following results:

$$X[k]=\sum_{n=0}^{N-1} x[n](-1)^n e^{\frac{-j2\pi kn}{N}}$$

$$X[k]^*=\sum_{n=0}^{N-1} x[n](-1)^n e^{\frac{-j2\pi k(-n)}{N}}$$

$$\mathit{Putting}\, m=-n\,,$$

$$X[k]^*=\sum_{m=0}^{N-1} x[-m](-1)^{-m} e^{\frac{-j2\pi km}{N}}=\mathscr{F}\left(x[-m](-1)^{-m}\right)$$

$$\mathscr{F}^{-1}(X[k]^*)=x[-n](-1)^{-n}$$

$$\mathit{After\ multiplying\ real\ part\ with}(-1)^n, \mathit{we\ get\ new\ signal}\ y[n]\ \mathit{as}$$

$$y[n]=x[-n]$$

Hence, we get y[n] to be mirrored version of x[n]. Same result can be extended to 2-D signals as well, since we can seperate the operations for column and row which will result in mirroring along horizontal axis and then along vertical axis passing through center of image.

Discussion:

- The kernel of the Fourier Transform has such structure that we can seperate it into operation along row and column individually. This fact allows us to first compute the transform along columns and then along rows, thereby reducing computational complexity from O(N^4) to O(N^2log(N)).
- All the images were found to be of low pass nature with the spectrum having a sharp white dot at the exact centre (DC component) and gradually becoming darker as we move away from the centre while following some pattern depending on the image.
- The phase spectrum of all the images were like a noisy image with sharp transition from black to white intensity levels.
- It was found that even if some values in the magnitude spectrum or phase spectrum went outside the range [0, 255], the generated image did not give any error and hence such values were automatically handled appropriately.
- The magnitude spectrum of *dip.tif* was found to be similar to *sinc* curve. The reason was that the characters in the image contained several rectangular white regions surrounded by black regions and rectangular pulse have *sinc* as Fourier Transform. Also, the phase spectrum was found to have some spiral kind of structure.
- It was found that when the $(-1)^{x+y}$ operation was neither applied before nor after the transforms, i.e, the operation was eliminated, the final image was still the same. This is evident from the proof shown above as well where we found that the $(-1)^{x+y}$ which was introduced before the transform, ultimately was cancelled after the transform.
- The operations mentioned in Q2 was applied to other images as well and was found to have the same effect of reflection about center on all the images.