Shri Ramdeobaba College of Engineering and Management, Nagpur

Department of Electronics Engineering

Digital Image Processing (ENT 355-3)

Name: Prajwal Pandurang Shette

Roll No: B1- 12

Experiment No: 10

Aim: The role of amplitude and phase of the Fourier transform in the digital image processing

Software Used: Python, Jupyter, OpenCV

Theory: The amplitude of an image represents the intensity of the different frequencies in the image. Therefore, it holds the geometrical structure of features in the image (i.e. changes in the spatial domain).
The phase on the other hand, represents the locations of these features (which helps our human eye to better comprehend the image).
A Signal is an electromagnetic field or an electric current to transmit data. There are various components of a signal such as frequency, amplitude, wavelength, phase, angular frequency and period from which it is described.
A periodic signal can be represented using the below sine function:

y = A sin(w*t + Q)

In which A represents the amplitude(in meter), w represents frequency(in hertz), t represents time period(in seconds) and Q represents phase(in radian) of the periodic signal.
The two major components frequency and phase of a periodic signal define the Phase Spectrum of that signal. The frequency components of the periodic signal are plotted in the horizontal axis and phase component of the periodic signal is plotted in the vertical axis.
In Python, the phase_spectrum() method in the pyplot module of Python matplotlib library plots the phase spectrum of a periodic signal. Below are some programs which demonstrate the use of phase_spectrum() method to visualize the phase spectrum of different periodic signals.

Code:

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('lake.jpg',0)
f = np.fft.fft2(img)
fshift1 = np.fft.fftshift(f)
magnitude_spectrumA = 20*np.log(np.abs(fshift1))
plt.subplot(121),plt.imshow(img, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(magnitude_spectrumA, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()
```

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
```

```
img=cv2.imread('lake.jpg')
img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
dft1 = np.fft.fft2(img)
dft_shift1 = np.fft.fftshift(dft1)
phase_spectrumA = np.angle(dft_shift1)

ax1 = plt.subplot(1,2,1)
ax1.imshow(img, cmap='gray')

ax2 = plt.subplot(1,2,2)
ax2.imshow(phase_spectrumA, cmap='gray')

plt.show()
```

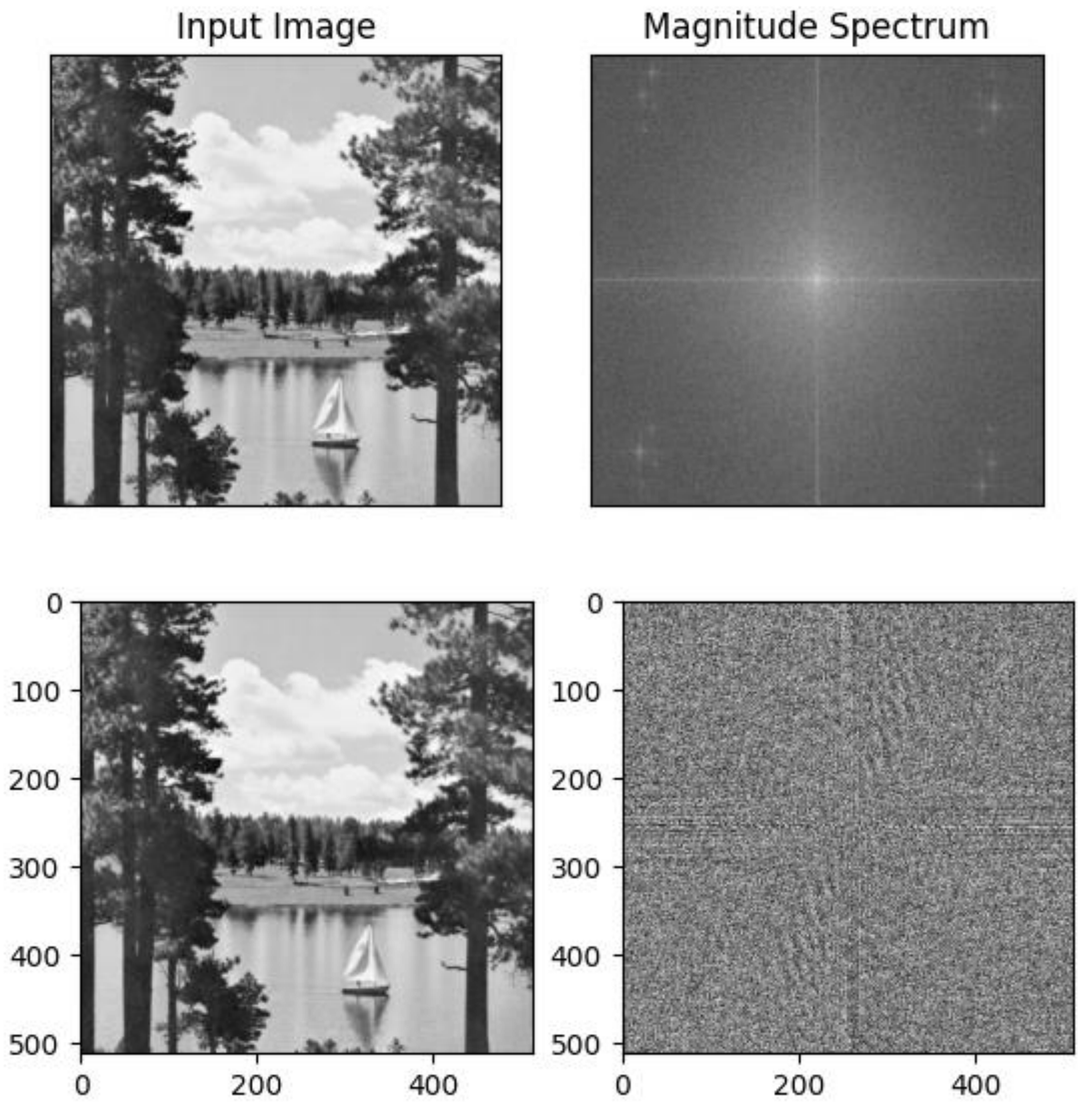
```
#image 2 phase and magnitude
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
img = cv.imread('plane.jpg',0)
f2 = np.fft.fft2(img)
fshift2 = np.fft.fftshift(f)
magnitude_spectrumB = 20*np.log(np.abs(fshift2))
plt.subplot(121),plt.imshow(img, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(magnitude_spectrumB, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()
```

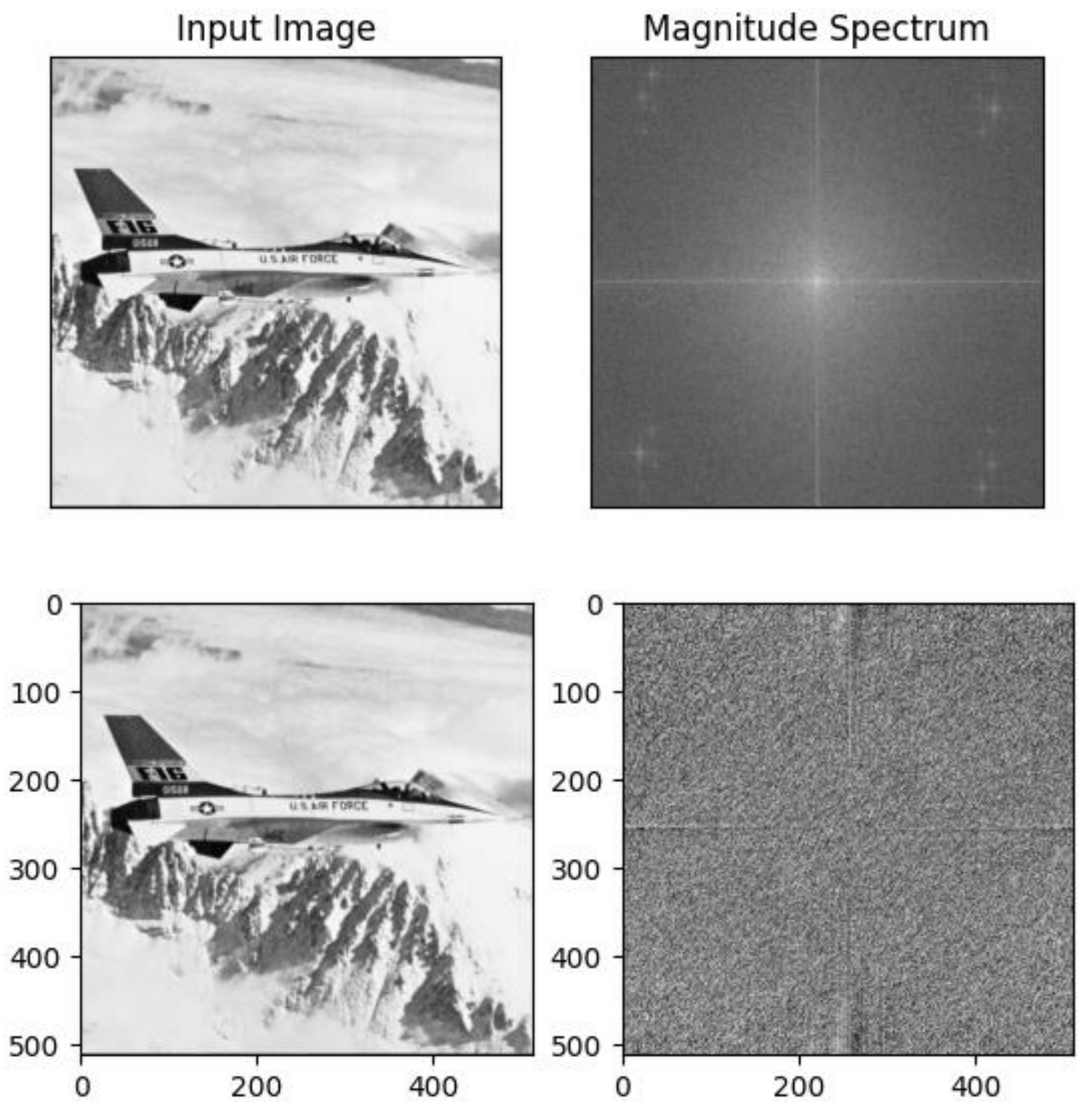
```

import numpy as np
import cv2
from matplotlib import pyplot as plt

img=cv2.imread('plane.jpg')
img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
dft2 = np.fft.fft2(img)
dft_shift2 = np.fft.fftshift(dft2)
phase_spectrumB = np.angle(dft_shift2)

ax1 = plt.subplot(1,2,1)
ax1.imshow(img, cmap='gray')
```

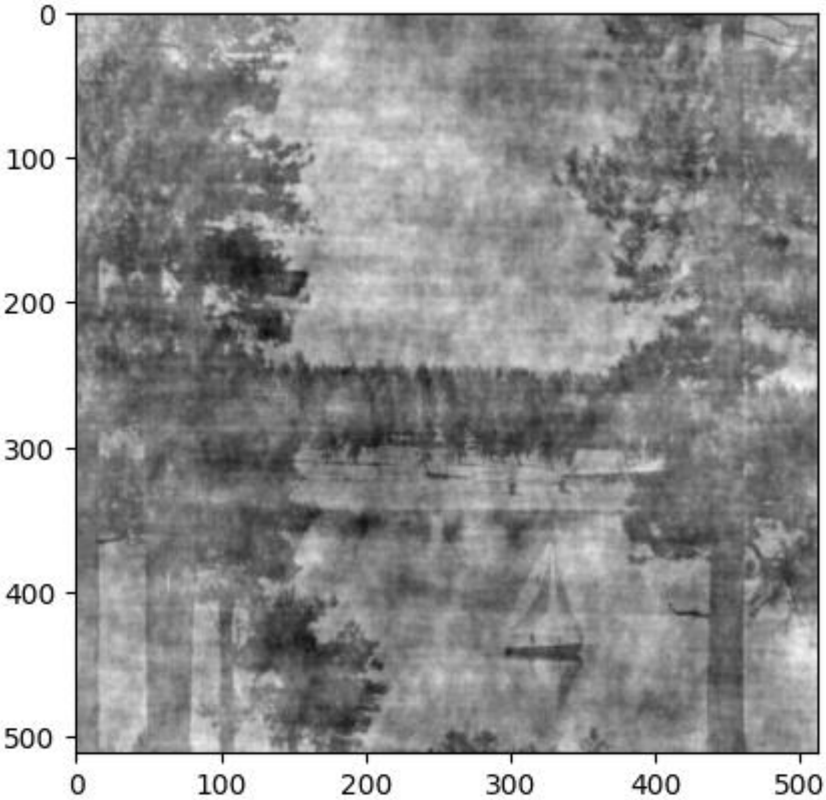
```python
ax2 = plt.subplot(1,2,2)
ax2.imshow(phase_spectrumB, cmap='gray')

plt.show()
```

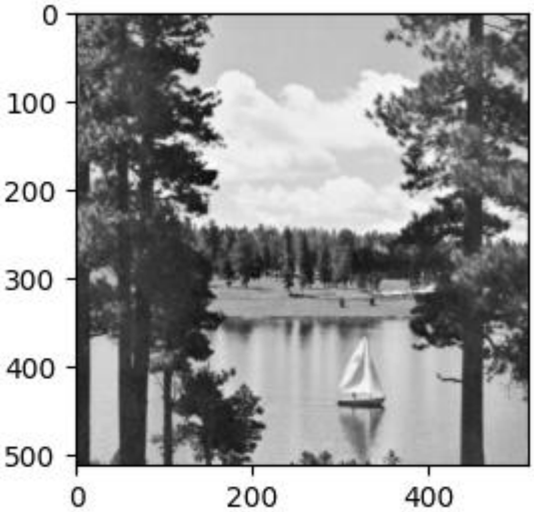
```python
combined = np.multiply(np.abs(f2), np.exp(1j*np.angle(f)))
imgCombined = np.real(np.fft.ifft2(combined))
import matplotlib.pyplot as plt
plt.imshow(imgCombined, cmap='gray')
```

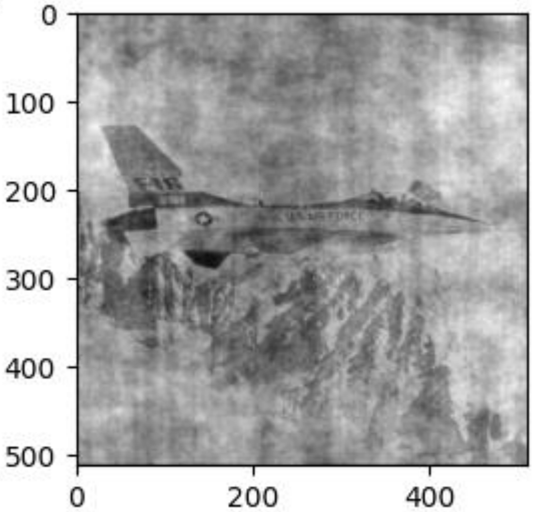
Output:

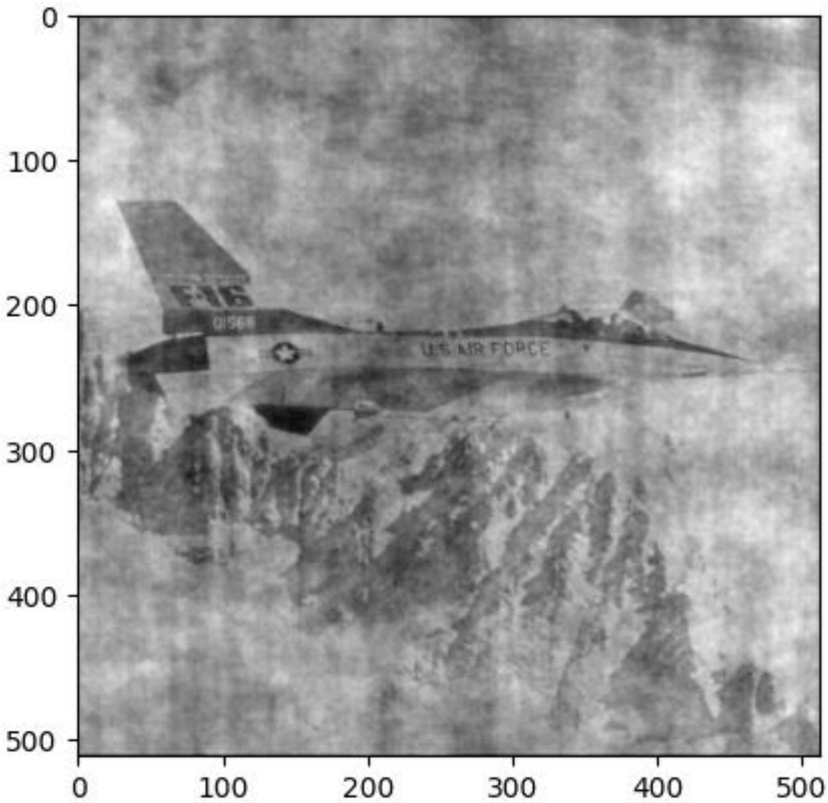



Input Image
Magnitude Spectrum
U.S. AIR FORCE
0
100
200
300
400
500
0
200
400
0
100
200
300
400
500
0
200
400


0
100
200
300
400
500
0
100
200
300
400
500


0
100
200
300
400
500
0
200
400


0
100
200
300
400
500
0
200
400



Observation & Conclusion:

We apply a Fourier transform, phase shift the transformed signal, and then perform the inverse Fourier transform to produce the phase shifted time domain signal.

Notice that the transforms are done with rfft() and irfft(), and that the phase shift is done by simply multiplying the transformed data by cmath.rect(1.,phase). The phase shift is equivalent to multiplying the complex transformed signal by exp(i * phase).

In the graphics, in the left panel, we show the original and shifted signals. The new signal is advanced by 90 degrees. In the right panel we show the power spectrum on the left axis. In this example we have power at a single frequency. Phase is graphed on the right axis. But again, since we have power at only one frequency, the phase spectrum shows noise everywhere else.