Semester I, Academic Year 2022–2023

Computational Optical Imaging

[Course Code: PYL759]

Submission: Assignment 1

Name : Mayand Dangi

Entry Number : 2019PH10637

Original Input Image

Code Snippets

Importing Package

```
import math
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
```

Loading the image and converting it to grayscale

```
img = Image.open('hw1.jpg')
img = img.convert('L')
img = img.resize((200,200))
g = np.asarray(img)
```

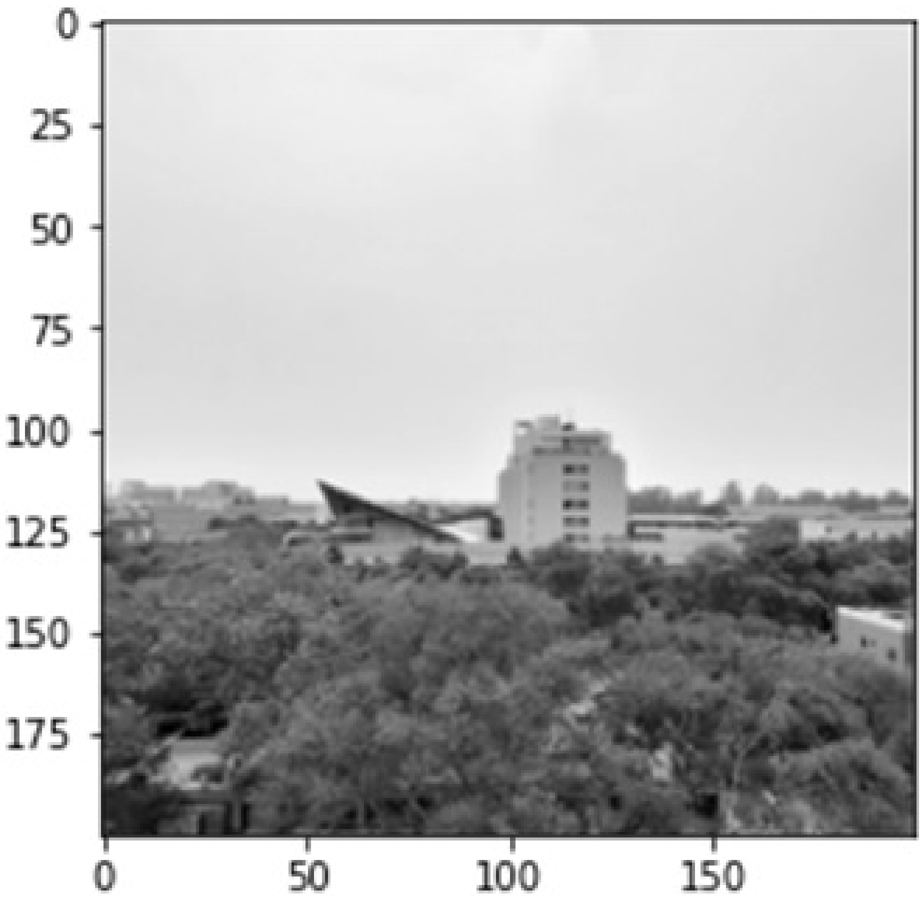

Figure 1: Gray scale image $g(x, y)$ of size 200×200 pixels.

Initializing parameters

```
N = img.size[0]          #size of image
p = 5e-6                 #pixel length (in m)
wavelength = 500e-9      #wavelength (in m)
z = 100e-06              #propagation distance (in m)

f0 = 1/(wavelength * math.sqrt(1+(2*z/(N*p))**2))

alpha = math.pi/(np.max(img_arr)*2)
```

Meshgrid in space and fourier planes

```
x,y = np.meshgrid(np.linspace(-N/2, N/2, N), np.linspace(-N/2, N/2, N))
x *= p
y *= p


fx,fy = np.meshgrid(np.linspace(-0.5, 0.5-1/N, N), np.linspace(-0.5, 0.5-1/N,
    N))
fx /= p
fy /= p
```

Defining circular mask of radius 100 pixels

```
circ = (abs(x)<=np.sqrt((100*p)**2-y**2))*(abs(y)<=np.sqrt((100*p)**2-x**2))
```

Defining the complex field $u = e^{i\alpha g} \times circ$

```
u = np.exp(1j*alpha*img_arr)*circ
```

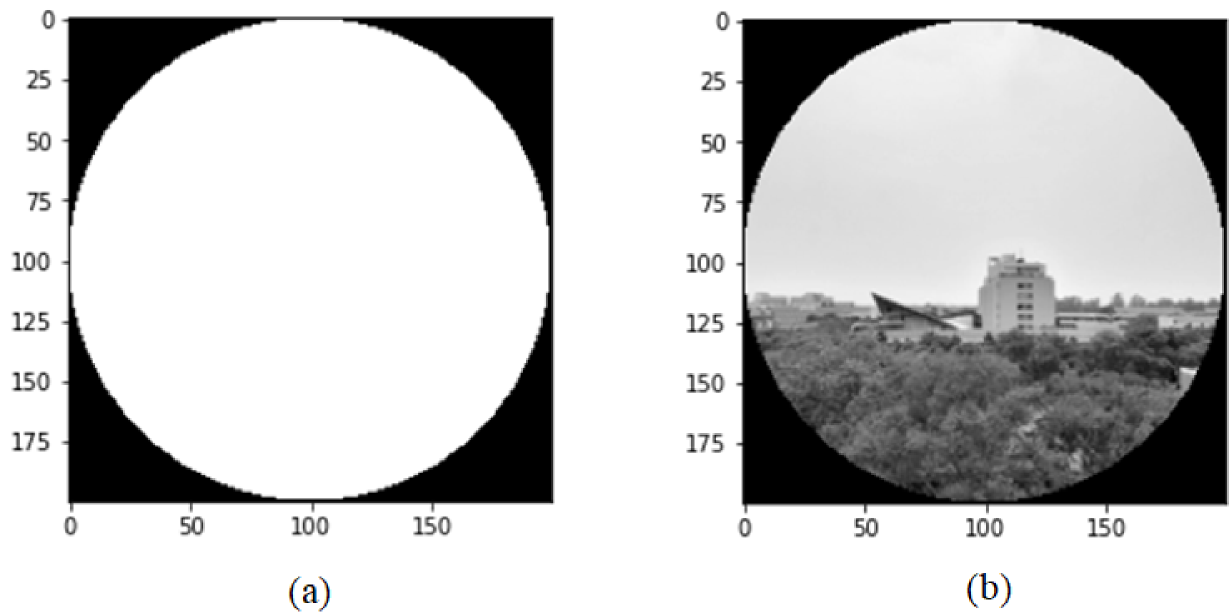

Figure 2: (a) Absolute value of field $u(x, y)$ (b) Phase of field $u(x, y)$

Propagating field using angular spectrum method

Function for propagating the field by distance z

```
'''
    @params initial_field: Field at z=0
    @params z: distance through field has to propagate
    @params wavelength: wavelength of field
    @params filtering: boolean
                            True -> Low Pass Filter will be applied
                            False-> No filtering
'''
def propagate_field(initial_field, z, wavelength, filtering):
    k = 2*math.pi/wavelength

    A = np.fft.fftshift(np.fft.fft2(np.fft.ifftshift(u)))

    alpha = np.sqrt(k**2 - 4*math.pi**2 *(fx**2+fy**2))
    H = np.exp(1j*alpha*z)

    # Field after traversing distance z
    if(filtering):
        LP = (abs(fx) <= np.sqrt(f0**2 - fy**2)) * (abs(fy) <= np.sqrt(f0**2 -
            fx**2))
```

```
        return np.fft.fftshift(np.fft.ifft2(np.fft.ifftshift(A*H*LP)))
    else:
        return np.fft.fftshift(np.fft.ifft2(np.fft.ifftshift(A*H)))
```

Thus, by propagating the field by z = $100\mu m$.

```
u_1 = propagate_field(u, z, wavelength, True)
```

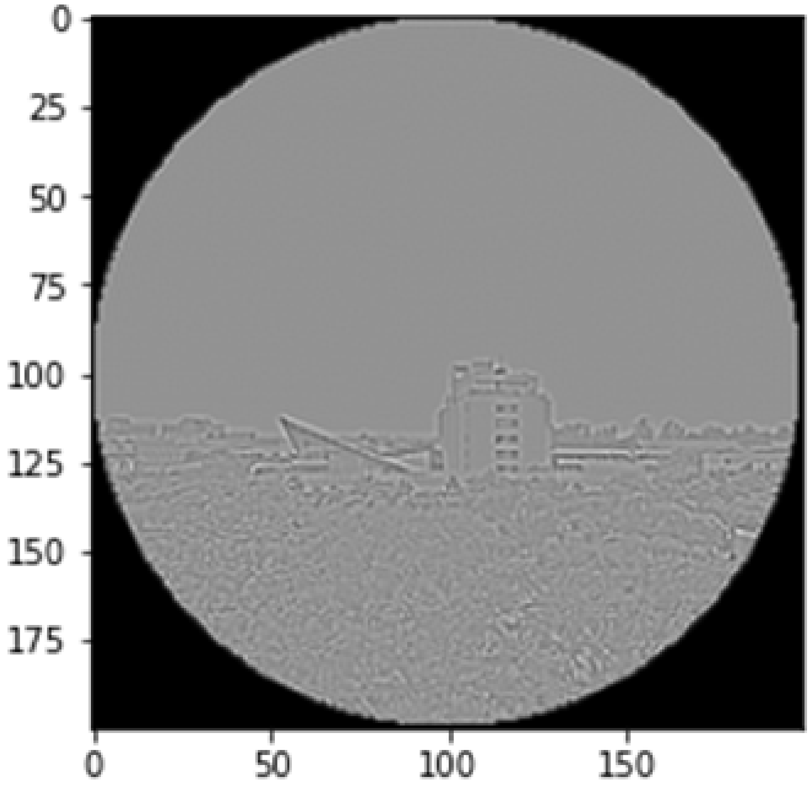

Figure 3: (a) $|u(x, y)|^2$ at $z = 100\mu m$

In Figure 3, we can observe the features similar to original image in the intensity of the propagated field. Since at z=0, our original image was completely in the phase term after the propagation it get appear in the amplitude also.

If we observe the image more closely then we can clearly see the edges of original image is appear in the propagated field intensity image. Thus our method might use in edge detection.