

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Birigui</p>	<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA Campus Birigui</p> <p><b>Bacharelado em Engenharia de Computação</b></p>	
<p><b>Disciplina:</b> Processamento Digital de Imagens</p>	<p>Filtragem Espacial</p>	
<p><b>Professor:</b> Prof. Dr. Murilo Vargas da Silva</p>	<p><b>Data:</b> 11/09/2023</p>	
<p><b>Nome do aluno:</b> Leonardo Reneres dos Santos</p>	<p><b>Prontuário:</b> BI3009131</p>	

### Questões:

- Implementar a Transformada de Fourier (Utilize a biblioteca de sua preferência)
- Implementar a Transformada Inversa de Fourier (Utilize a biblioteca de sua preferência)
- Plotar o espectro e fase.
- Comparar os resultados com ImageJ.
- Plotar o espectro 3D (Pesquisar formas de visualização 3D em Python)
  - Utilizar as imagens disponibilizadas na aula ([Images\\_fourier.rar](#))
  - Criar uma imagem fundo branco e um quadrado simulando a função SINC

### Respostas:

#### código:

```
import matplotlib.pyplot as plt

import cv2 as cv

import numpy as np

from PIL import Image

from mpl_toolkits import mplot3d

def inverse_fourier_transform(fshift):
```

```

    f_ishift = np.fft.ifftshift(fshift)

    im1_back = np.fft.ifft2(f_ishift)

    im1_back = np.abs(im1_back)

    return im1_back

def plt3d(im1):

    fig = plt.figure()

    ax = fig.add_subplot(111, projection='3d')

    x = np.arange(0, im1.shape[0], 1)

    y = np.arange(0, im1.shape[1], 1)

    X,Y = np.meshgrid(x, y)

    z = im1[X,Y]

    ax.plot_surface(X, Y, z, cmap='viridis', edgecolor='none')

    ax.set_xlabel('x')

    ax.set_ylabel('y')

    plt.show()

def main():

    while(True):

        print("Selecione a imagem: 1 , 2, 3, 4 ou 5")

        im = input()

        if im == '1':

            im1 =cv.imread("./len_periodic_noise.png",
cv.IMREAD_GRAYSCALE)

            elif im == '2':

                im1 = cv.imread('./periodic_noise.png',
cv.IMREAD_GRAYSCALE)

            elif im == '3':

```

```

        im1 = cv.imread('./car.tif', cv.IMREAD_GRAYSCALE)

    elif im == '4':

        im1 = cv.imread('./sinc.png', cv.IMREAD_GRAYSCALE)

    elif im == '5':

        im1 =
cv.imread('./newspaper_shot_woman.tif',cv.IMREAD_GRAYSCALE )

    elif im == '6':

        break

    else:

        im1 = cv.imread("./len_periodic_noise.png",
cv.IMREAD_GRAYSCALE)

        while(True):

            k = input("1 para sair, qualquer outra tecla para
continuar:")

            if(k == '1'):

                break

            else:

                assert im1 is not None, "file could not be read, check
with os.path.exists()"

                f = np.fft.fft2(im1)

                fshift = np.fft.fftshift(f)

                magnitude_spectrum = 20*np.log(np.abs(fshift))

                fase = np.angle(fshift)

                plt.subplot(141),plt.imshow(im1, cmap = 'gray')

                plt.title('Input Image'), plt.xticks([],
plt.yticks([])

```

```

plt.subplot(142),plt.imshow(magnitude_spectrum, cmap =
'gray')

plt.title('Magnitude Spectrum'), plt.xticks([]),
plt.yticks([])

plt.subplot(143),plt.imshow(fase, cmap = 'gray')

plt.title('Fase'), plt.xticks([]), plt.yticks([])

plt.subplot(144),plt.imshow(inverse_fourier_transform(fshift), cmap =
'gray')

plt.title('Image after HPF'), plt.xticks([]),
plt.yticks([])

plt.show()

#plt3d(im1)

plt3d(magnitude_spectrum)

if __name__ == "__main__":

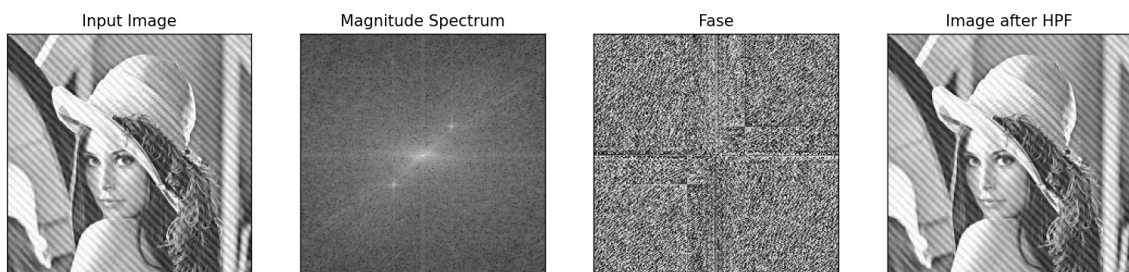
    main()

```

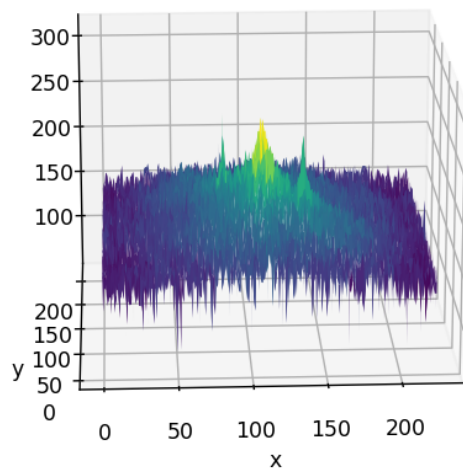
**prints:**

len\_periodic\_noise

imagens

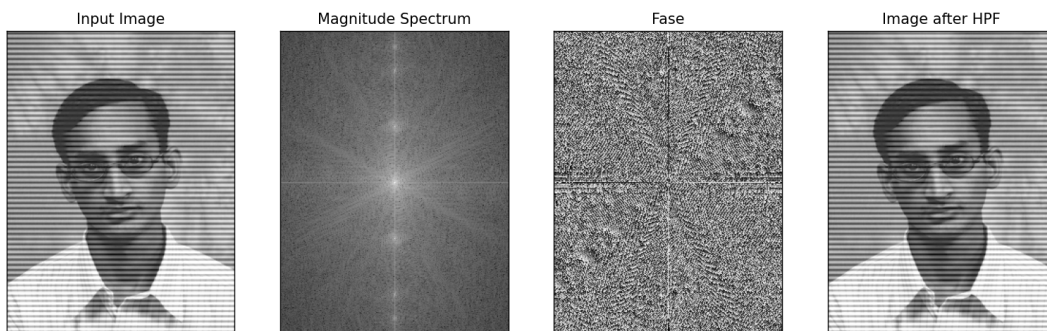


3D

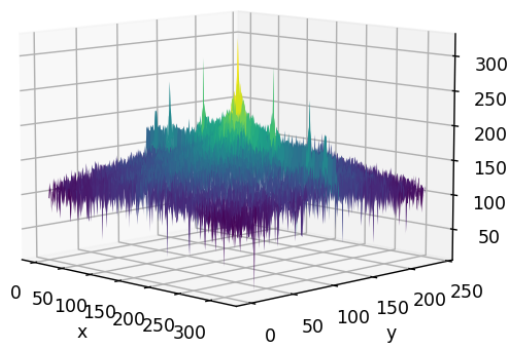


periodic\_noise

imagens

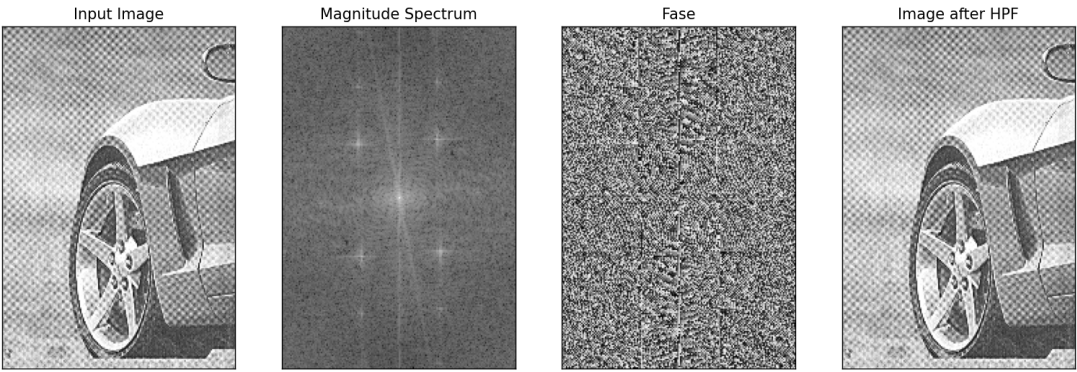


3D

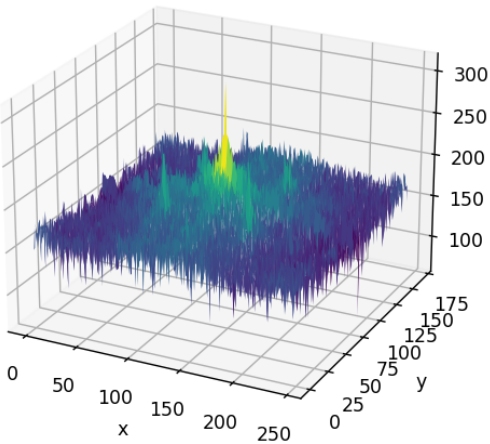


car

imagens

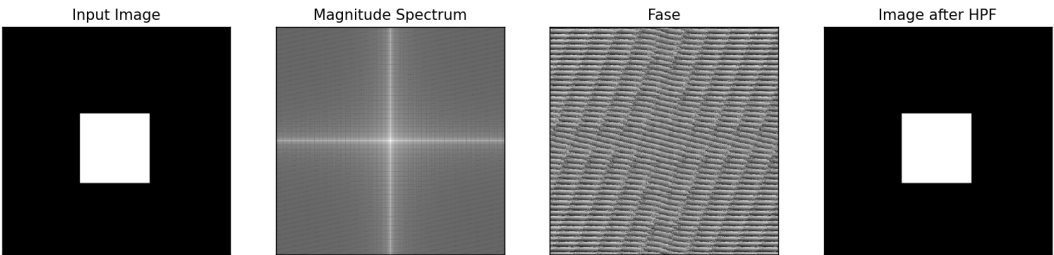


3D

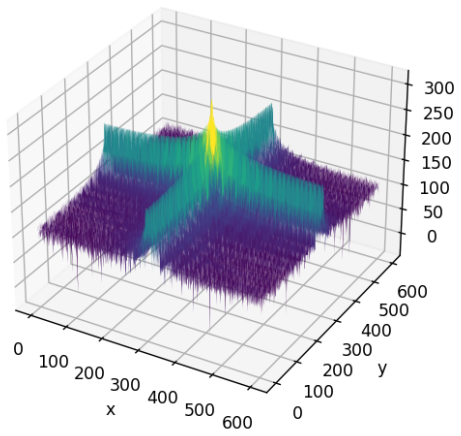


sinc

imagens

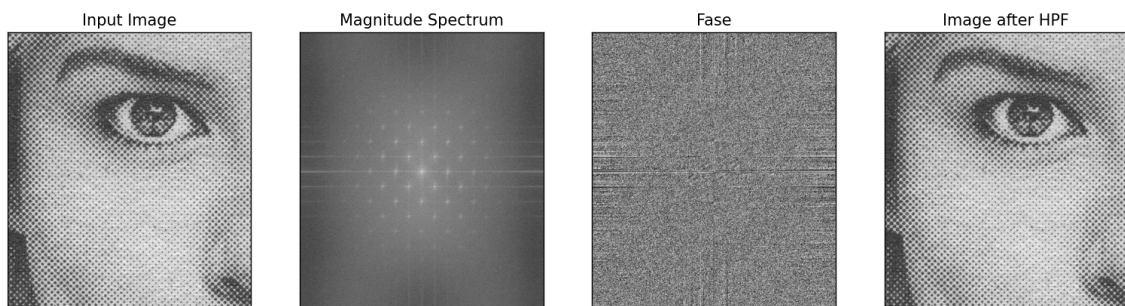


3D



newspaper

imagens



3D

