



**CAMPUS BIRIGUI**

**ATIVIDADES - TRANSFORMADA DE FOURIER**

**JOÃO EDUARDO SANTO FARIAS**

**PROF. DR. MURILO VARGES DA SILVA**

**SETEMBRO DE 2023**

- 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

A Transformada de Fourier é uma ferramenta matemática fundamental que desempenha um papel essencial em diversas áreas, como processamento de sinais, processamento de imagens, análise de espectro e muitas outras aplicações em ciência e engenharia. Em Python, a Transformada de Fourier pode ser implementada com facilidade, graças ao suporte de bibliotecas como NumPy e SciPy.

Essa técnica permite a decomposição de um sinal ou imagem em suas componentes de frequência, revelando informações cruciais sobre sua composição espectral. Ela é amplamente utilizada em áreas como processamento de áudio, análise de séries temporais, processamento de imagens médicas, comunicações, entre outras.

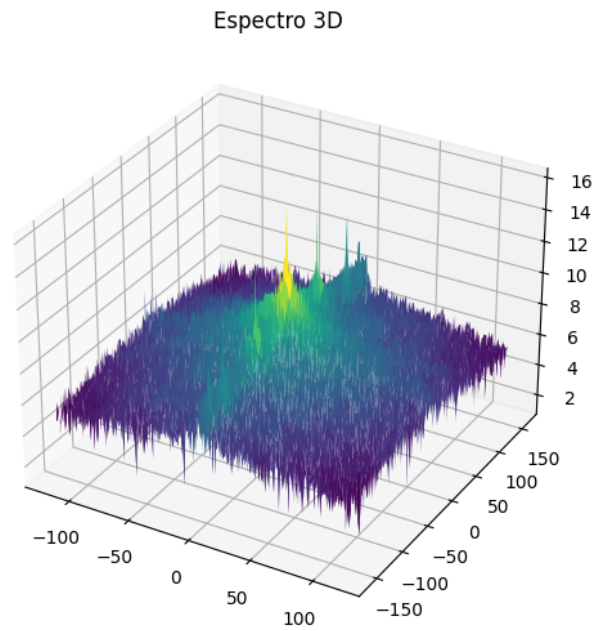
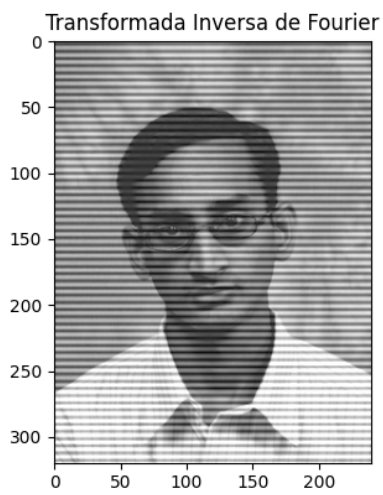
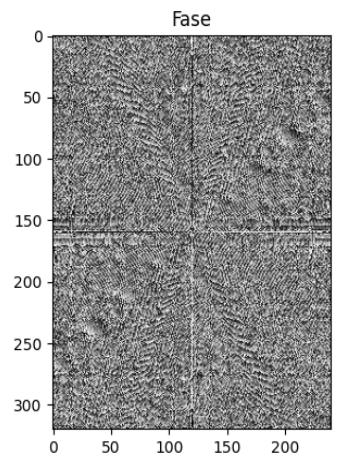
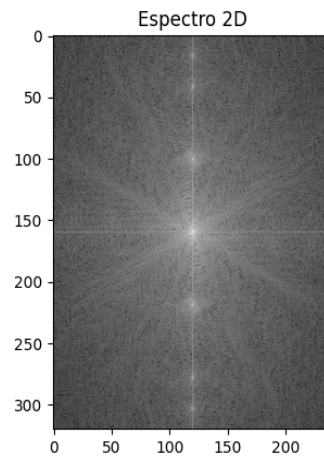
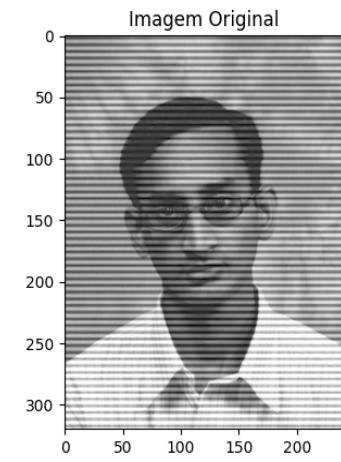
Primeiro, uma imagem é carregada a partir do caminho especificado e convertida para escala de cinza, se necessário. A imagem original é então exibida em um gráfico 2D em escala de cinza.

Em seguida, a Transformada de Fourier 2D é aplicada à imagem original usando a função `fft2` da biblioteca SciPy. Isso calcula o espectro de frequência da imagem. O logaritmo do valor absoluto do espectro é calculado para realçar os detalhes e evitar valores muito altos. O resultado é exibido como um gráfico 2D do espectro.

A fase da Transformada de Fourier é calculada usando `np.angle` e exibida como um terceiro gráfico 2D, representando a fase da imagem. A Transformada Inversa de Fourier é calculada com `ifft2` para recuperar a imagem original a partir do espectro de frequência.

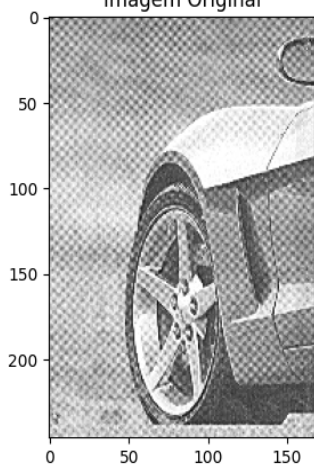
Finalmente, o código também inclui a plotagem do espectro em 3D usando `matplotlib`, onde o espectro é projetado em um espaço tridimensional para visualizar sua estrutura de frequência. Isso é útil para analisar as componentes de frequência da imagem.

## PERIODIC\_NOISE

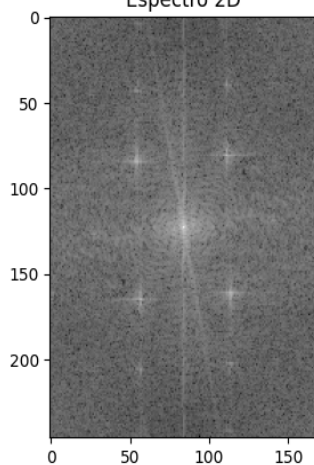


## CAR

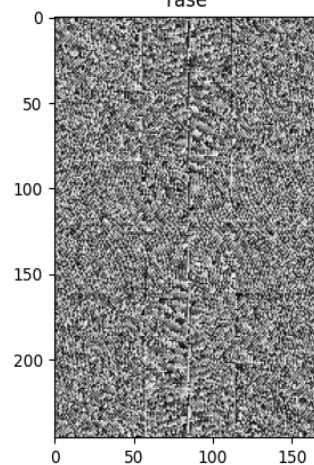
Imagem Original



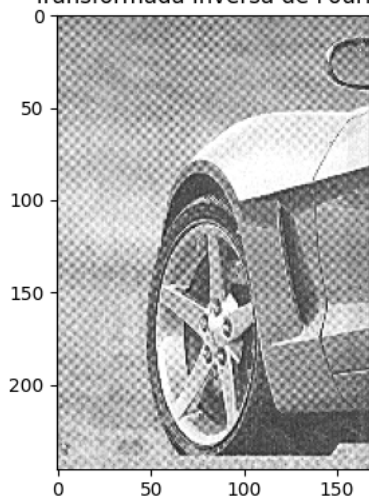
Espectro 2D



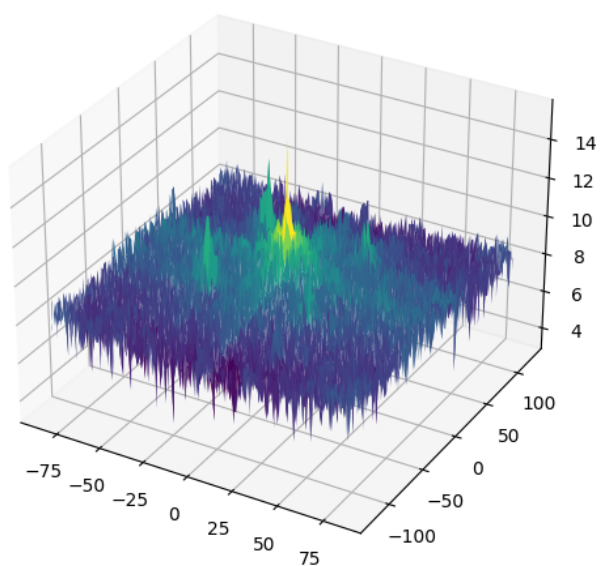
Fase



Transformada Inversa de Fourier

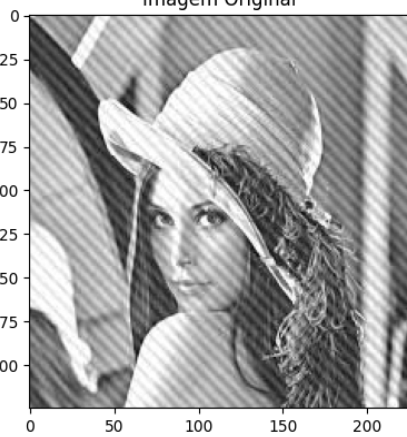


Espectro 3D

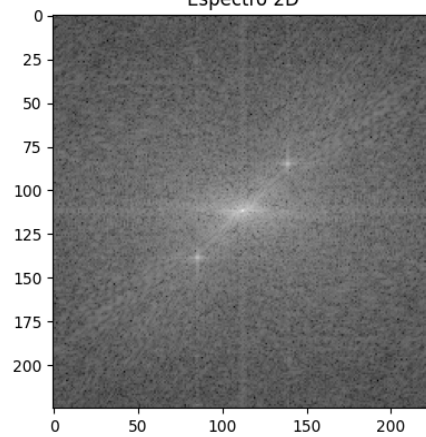


# LEN\_PERIODIC\_NOISE

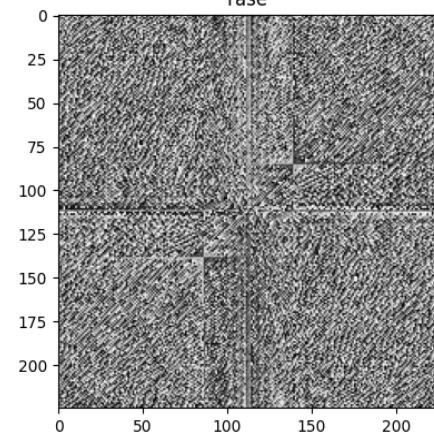
Imagem Original



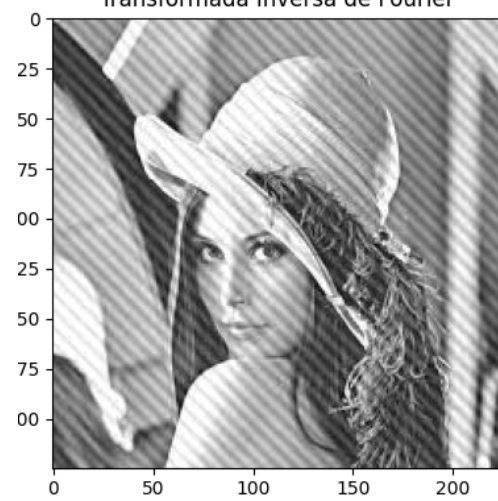
Espectro 2D



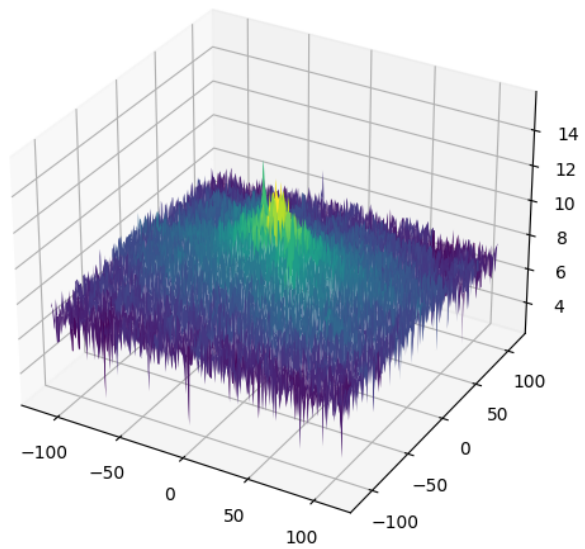
Fase



Transformada Inversa de Fourier



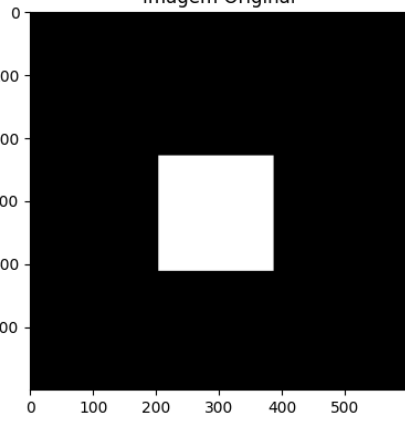
Espectro 3D



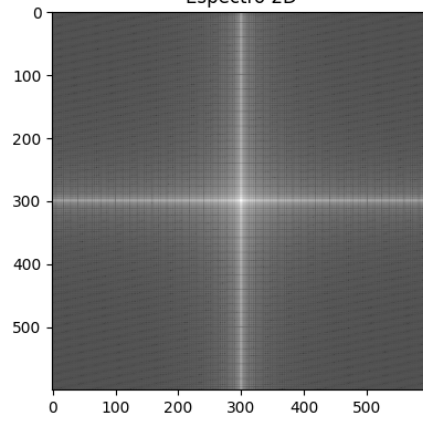


# SINC

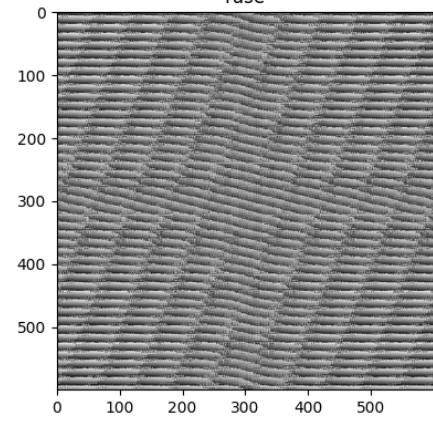
Imagem Original



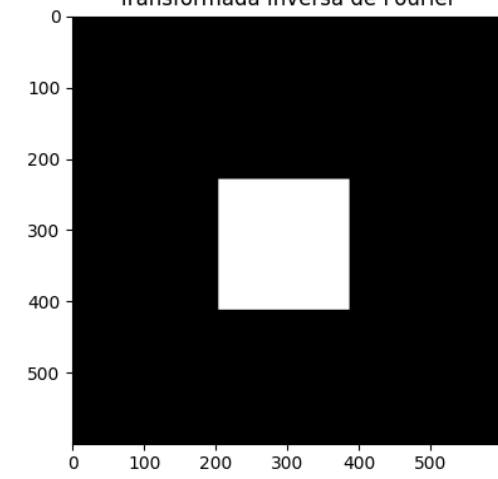
Espectro 2D



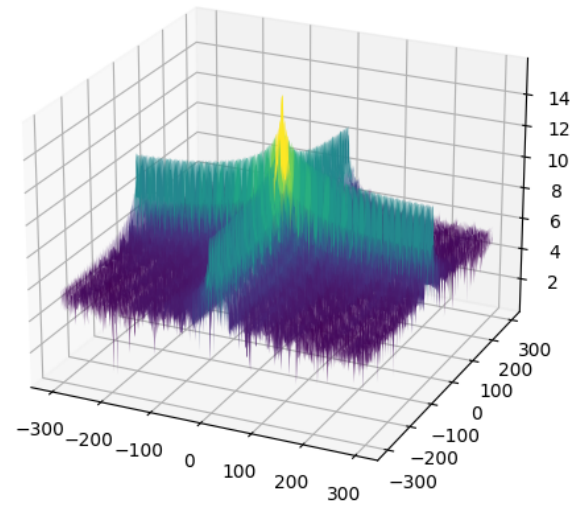
Fase



Transformada Inversa de Fourier

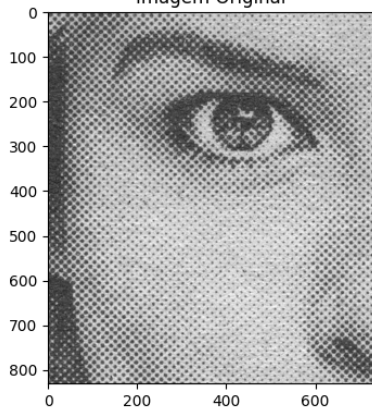


Espectro 3D

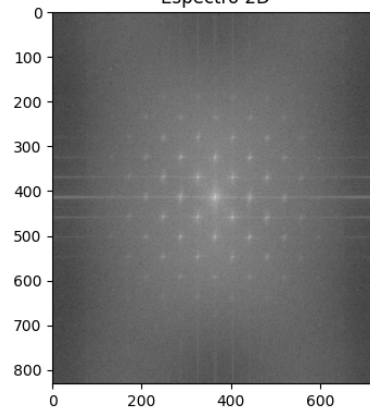


# NEWSPAPER\_SHOT\_WOMAN

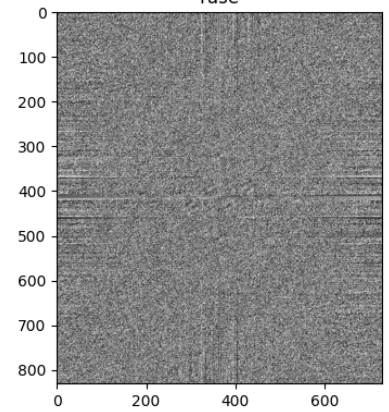
Imagem Original



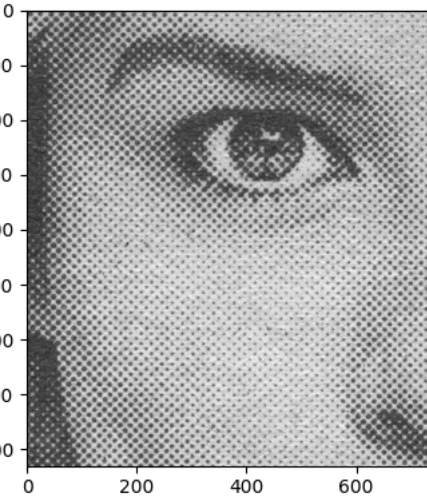
Espectro 2D



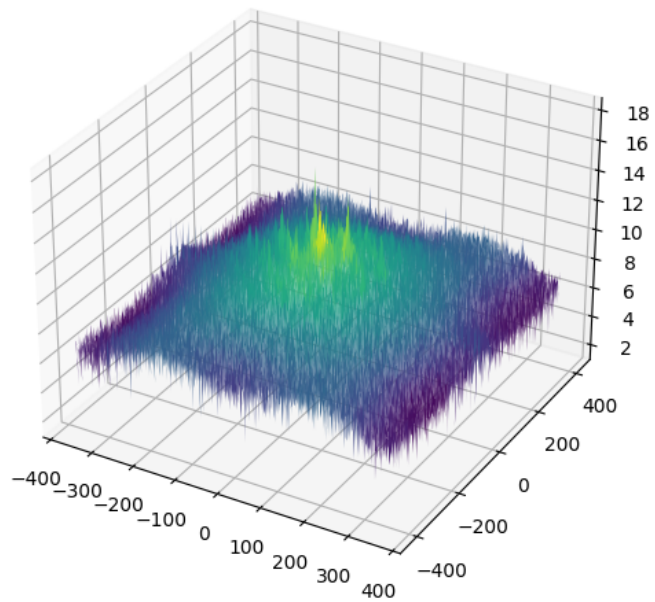
Fase



Transformada Inversa de Fourier



Espectro 3D



## Código

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.fft import fft2, ifft2, fftshift
from PIL import Image

# Insira o caminho da sua própria imagem
imagem_caminho = 'LEN_PERIODIC_NOISE.png'

# Carregar sua própria imagem
imagem = Image.open(imagem_caminho)
imagem = imagem.convert('L') # Converter para escala de cinza, se
necessário

# Converter a imagem em uma matriz NumPy
image = np.array(imagem)

# Calcula a Transformada de Fourier 2D
fourier_transform = fft2(image)

# Calcula a Transformada Inversa de Fourier
inverse_fourier_transform = ifft2(fourier_transform)

# Calcula o espectro e a fase
spectrum = np.abs(fftshift(fourier_transform))
phase = np.angle(fftshift(fourier_transform))

# Plotar a imagem original
plt.figure(figsize=(12, 5))
plt.subplot(231)
plt.title("Imagem Original")
plt.imshow(image, cmap='gray')

# Plotar o espectro em 2D
plt.subplot(232)
plt.title("Espectro 2D")
plt.imshow(np.log(spectrum + 1), cmap='gray')

# Plotar a fase
plt.subplot(233)
plt.title("Fase")
plt.imshow(phase, cmap='gray')
```



```
# Plotar a Transformada Inversa de Fourier
plt.subplot(234)
plt.title("Transformada Inversa de Fourier")
plt.imshow(np.real(inverse_fourier_transform), cmap='gray')

# Plotar o espectro em 3D
plt.figure(figsize=(8, 6))
ax = plt.axes(projection='3d')
ax.set_title("Espectro 3D")
x = np.linspace(-spectrum.shape[1] // 2, spectrum.shape[1] // 2 - 1,
spectrum.shape[1])
y = np.linspace(-spectrum.shape[0] // 2, spectrum.shape[0] // 2 - 1,
spectrum.shape[0])
X, Y = np.meshgrid(x, y)
ax.plot_surface(X, Y, np.log(spectrum + 1), cmap='viridis')
plt.show()
```