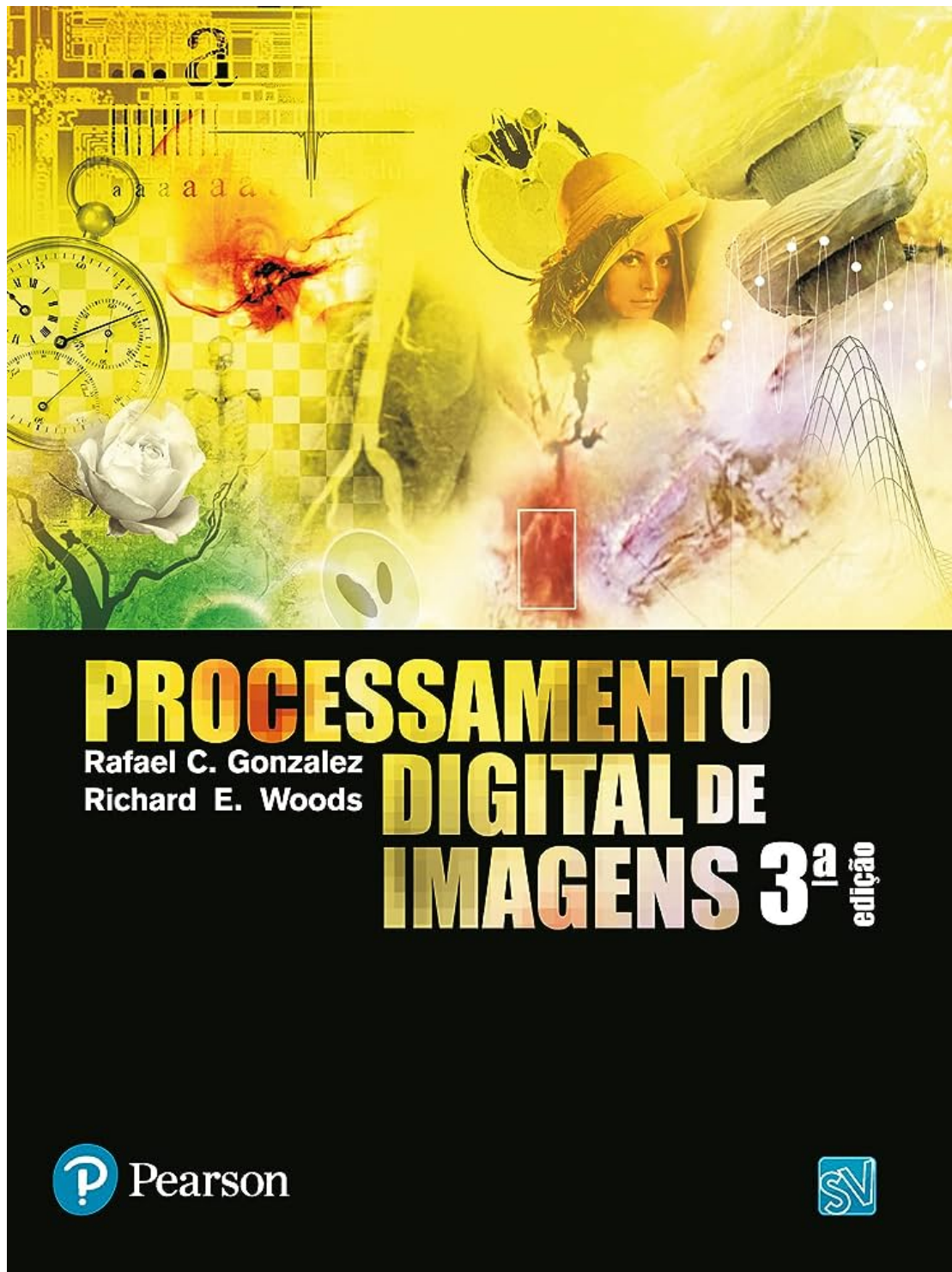


# PDIE8 - Processamento Digital de Imagem



Repositorio dos Códigos feitos em aula da Disciplina

Feito por: Vinicius de Souza Santos

Lecionado por: Murilo Varges

# Aula 7 - 11/09 - Domínio da Frequência

- Domínio da frequência
- Transformada de Fourier
- Filtragem no domínio da frequência

## Atividades - Transformada de Fourier

- Implementar a Transformada de Fourier (Utilize a biblioteca de sua preferência)
- Plotar o espectro e fase.
- 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
- Plotar Original

## Plotando Espectro e Fase

Importando Bibliotecas

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt
import cv2
```

Imagem Lena

```
In [ ]: imgLen = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de I
```

Processamento para cada imagem

```
In [ ]: # Função para calcular a Transformada de Fourier 2D e plotar a imagem original,
def calcular_e_plotar_fft(imagem, titulo):
    # Calcule a Transformada de Fourier 2D da imagem
    transformada_fourier = np.fft.fft2(imagem)

    # Calcule o espectro de magnitude
    espectro_magnitude = np.abs(transformada_fourier)

    # Calcule a fase
    fase = np.angle(transformada_fourier)

    # Plote a imagem original, o espectro de magnitude e a fase
    plt.figure(figsize=(18, 4))

    plt.subplot(1, 3, 1)
    plt.imshow(imagem, cmap='gray')
    plt.title('Imagem Original')

    plt.subplot(1, 3, 2)
```

```

plt.imshow(np.log(1 + espectro_magnitude), cmap='gray')
plt.title('Espectro de Magnitude')
plt.colorbar()

plt.subplot(1, 3, 3)
plt.imshow(fase, cmap='gray')
plt.title('Fase')
plt.colorbar()

plt.tight_layout()
plt.show()

# Chamada da função para cada imagem
calcular_e_plotar_fft(imgLen, "imgLen")

```

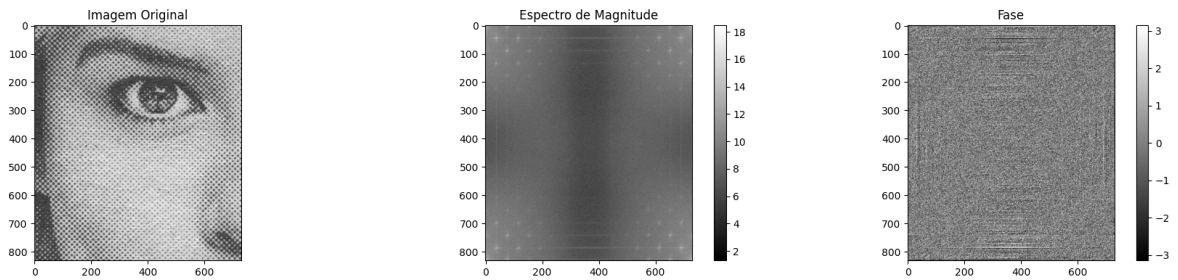


Imagem Car

```
In [ ]: imgCar = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de I
```

```

In [ ]: # Função para calcular a Transformada de Fourier 2D e plotar a imagem original,
def calcular_e_plotar_fft(imagem, titulo):
    # Calcule a Transformada de Fourier 2D da imagem
    transformada_fourier = np.fft.fft2(imagem)

    # Calcule o espectro de magnitude
    espectro_magnitude = np.abs(transformada_fourier)

    # Calcule a fase
    fase = np.angle(transformada_fourier)

    # Plote a imagem original, o espectro de magnitude e a fase
    plt.figure(figsize=(18, 4))

    plt.subplot(1, 3, 1)
    plt.imshow(imagem, cmap='gray')
    plt.title('Imagem Original')

    plt.subplot(1, 3, 2)
    plt.imshow(np.log(1 + espectro_magnitude), cmap='gray')
    plt.title('Espectro de Magnitude')
    plt.colorbar()

    plt.subplot(1, 3, 3)
    plt.imshow(fase, cmap='gray')
    plt.title('Fase')
    plt.colorbar()

    plt.tight_layout()
    plt.show()

```

```
# Chamada da função para cada imagem
calcular_e_plotar_fft(imgCar, "imgCar")
```

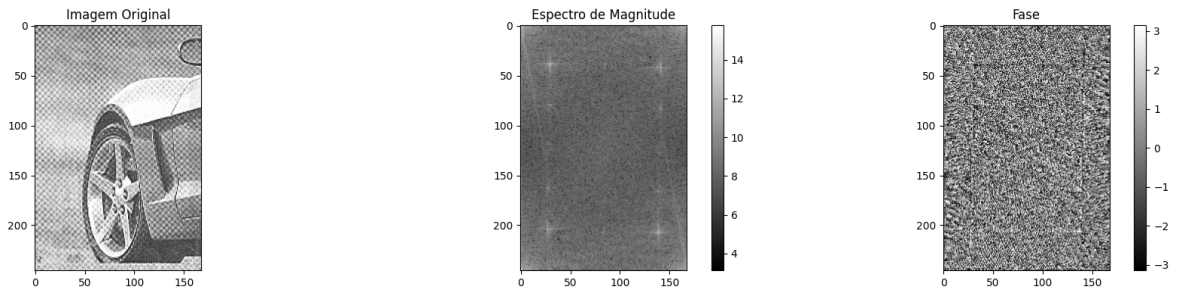


Imagem newspaper\_shot\_woman

```
In [ ]: imgnewspaper_shot_woman = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processam
```

```
In [ ]: # Função para calcular a Transformada de Fourier 2D e plotar a imagem original,
def calcular_e_plotar_fft(imagem, titulo):
    # Calcule a Transformada de Fourier 2D da imagem
    transformada_fourier = np.fft.fft2(imagem)

    # Calcule o espectro de magnitude
    espectro_magnitude = np.abs(transformada_fourier)

    # Calcule a fase
    fase = np.angle(transformada_fourier)

    # Plote a imagem original, o espectro de magnitude e a fase
    plt.figure(figsize=(18, 4))

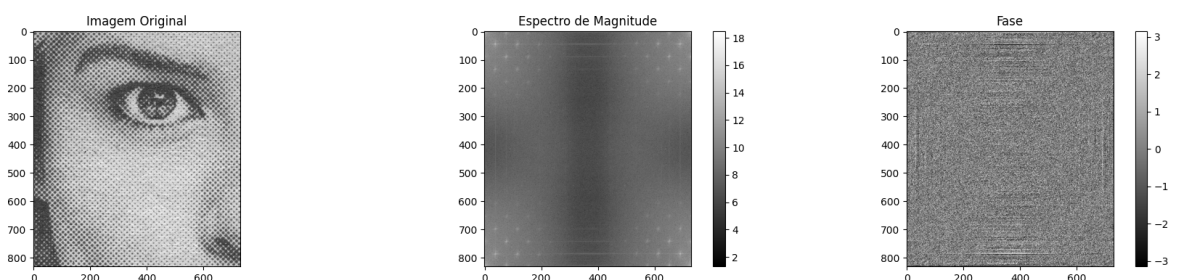
    plt.subplot(1, 3, 1)
    plt.imshow(imagem, cmap='gray')
    plt.title('Imagem Original')

    plt.subplot(1, 3, 2)
    plt.imshow(np.log(1 + espectro_magnitude), cmap='gray')
    plt.title('Espectro de Magnitude')
    plt.colorbar()

    plt.subplot(1, 3, 3)
    plt.imshow(fase, cmap='gray')
    plt.title('Fase')
    plt.colorbar()

    plt.tight_layout()
    plt.show()

# Chamada da função para cada imagem
calcular_e_plotar_fft(imgnewspaper_shot_woman, "imgnewspaper_shot_woman")
```



## Imagem periodic\_noise

```
In [ ]: imagperiodic_noise = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processamento
```

```
In [ ]: # Função para calcular a Transformada de Fourier 2D e plotar a imagem original,
def calcular_e_plotar_fft(imagem, titulo):
    # Calcule a Transformada de Fourier 2D da imagem
    transformada_fourier = np.fft.fft2(imagem)

    # Calcule o espectro de magnitude
    espectro_magnitude = np.abs(transformada_fourier)

    # Calcule a fase
    fase = np.angle(transformada_fourier)

    # Plote a imagem original, o espectro de magnitude e a fase
    plt.figure(figsize=(18, 4))

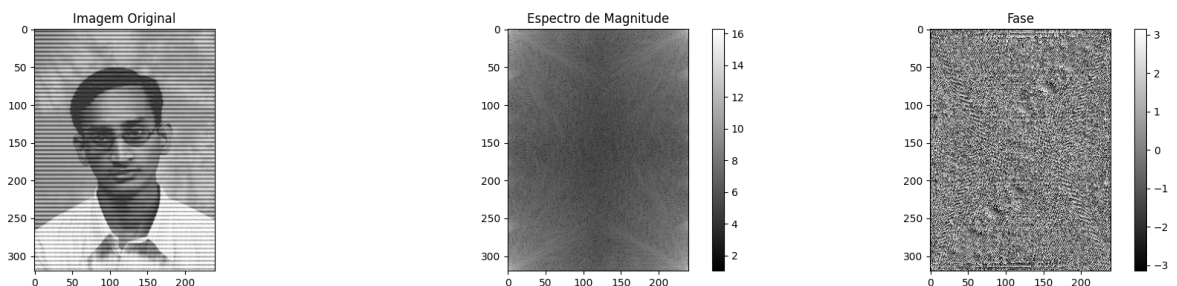
    plt.subplot(1, 3, 1)
    plt.imshow(imagem, cmap='gray')
    plt.title('Imagem Original')

    plt.subplot(1, 3, 2)
    plt.imshow(np.log(1 + espectro_magnitude), cmap='gray')
    plt.title('Espectro de Magnitude')
    plt.colorbar()

    plt.subplot(1, 3, 3)
    plt.imshow(fase, cmap='gray')
    plt.title('Fase')
    plt.colorbar()

    plt.tight_layout()
    plt.show()

    # Chamada da função para cada imagem
    calcular_e_plotar_fft(imagperiodic_noise, "imagperiodic_noise")
```



## Imagem sinc

```
In [ ]: imgsinc = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de
```

```
In [ ]: # Função para calcular a Transformada de Fourier 2D e plotar a imagem original,
def calcular_e_plotar_fft(imagem, titulo):
    # Calcule a Transformada de Fourier 2D da imagem
    transformada_fourier = np.fft.fft2(imagem)

    # Calcule o espectro de magnitude
    espectro_magnitude = np.abs(transformada_fourier)
```



```

# Calcule a fase
fase = np.angle(transformada_fourier)

# Plote a imagem original, o espectro de magnitude e a fase
plt.figure(figsize=(18, 4))

plt.subplot(1, 3, 1)
plt.imshow(imagem, cmap='gray')
plt.title('Imagem Original')

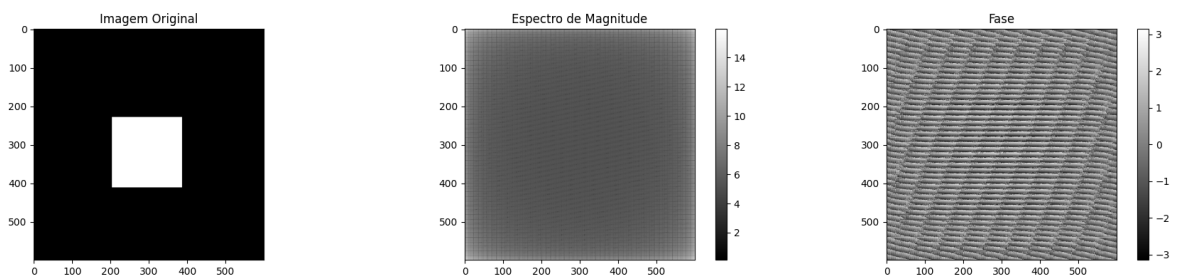
plt.subplot(1, 3, 2)
plt.imshow(np.log(1 + espectro_magnitude), cmap='gray')
plt.title('Espectro de Magnitude')
plt.colorbar()

plt.subplot(1, 3, 3)
plt.imshow(fase, cmap='gray')
plt.title('Fase')
plt.colorbar()

plt.tight_layout()
plt.show()

# Chamada da função para cada imagem
calcular_e_plotar_fft(imagsinc, "imagsinc")

```



## 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

Importando Bibliotecas

```

In [ ]: import numpy as np
import matplotlib.pyplot as plt
import cv2
from mpl_toolkits.mplot3d import Axes3D

```

Carregando as imagens

```

In [ ]: imgCar = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem/Aulas/Aula 7/Aula 7.html")
imgLen = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem/Aulas/Aula 7/Aula 7.html")
imgNS = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem/Aulas/Aula 7/Aula 7.html")

```

```
imgPeriodic = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem/imgPeriodic.png")
imgSinc = cv2.imread("/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem/imgSinc.png")
```

Função para calcular e plotar o espectro 3D

```
In [ ]: def plotar_espectro_3D(imagem, titulo):
        # Calcule a Transformada de Fourier 2D da imagem
        transformada_fourier = np.fft.fft2(imagem)

        # Calcule o espectro 2D
        espectro_2D = np.fft.fftshift(transformada_fourier)

        # Calcule o espectro de magnitude
        espectro_magnitude = np.abs(espectro_2D)

        # Crie uma grade de coordenadas para o espectro 3D
        x = np.fft.fftshift(np.fft.fftfreq(imagem.shape[1]))
        y = np.fft.fftshift(np.fft.fftfreq(imagem.shape[0]))
        X, Y = np.meshgrid(x, y)

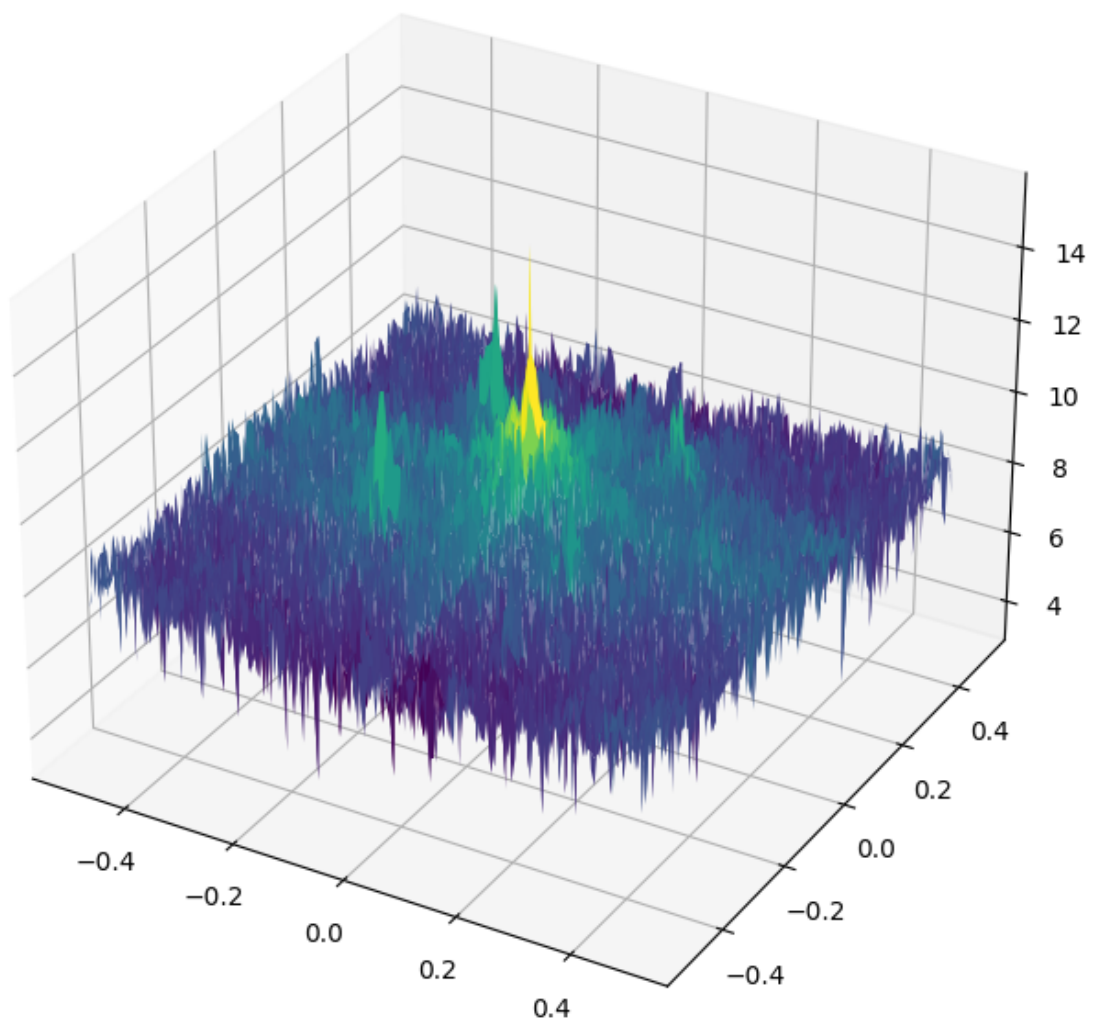
        # Plote o espectro 3D
        fig = plt.figure(figsize=(10, 8))
        ax = fig.add_subplot(111, projection='3d')
        ax.set_title('Espectro 3D - ' + titulo)
        ax.plot_surface(X, Y, np.log(1 + espectro_magnitude), cmap='viridis')

        plt.show()
```

Plotando Imagem Carro

```
In [ ]: plotar_espectro_3D(imgCar, "imgCar")
```

## Espectro 3D - imgCar

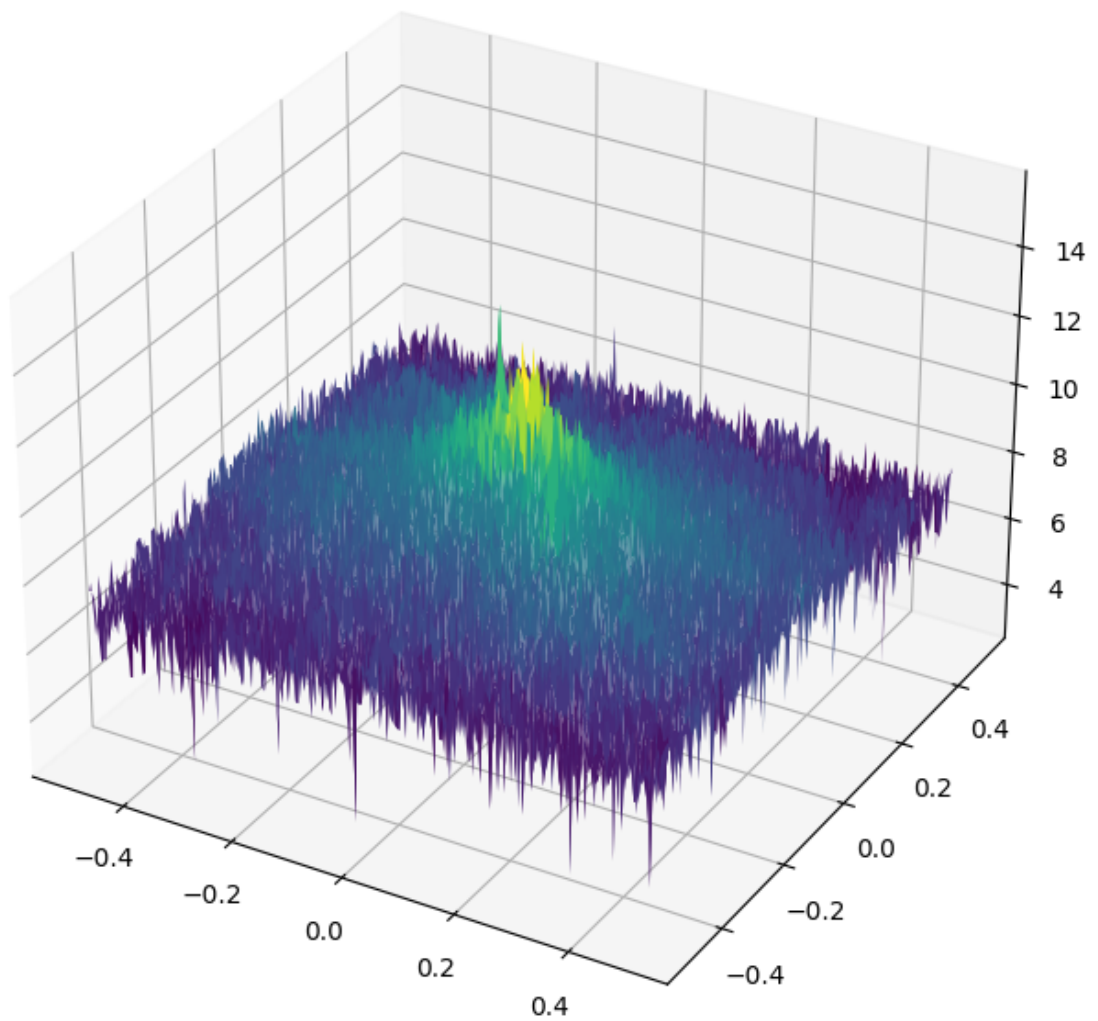


## Plotando Imagem Lena

```
In [ ]: plotar_espectro_3D(imgLen, "imgLen")
```



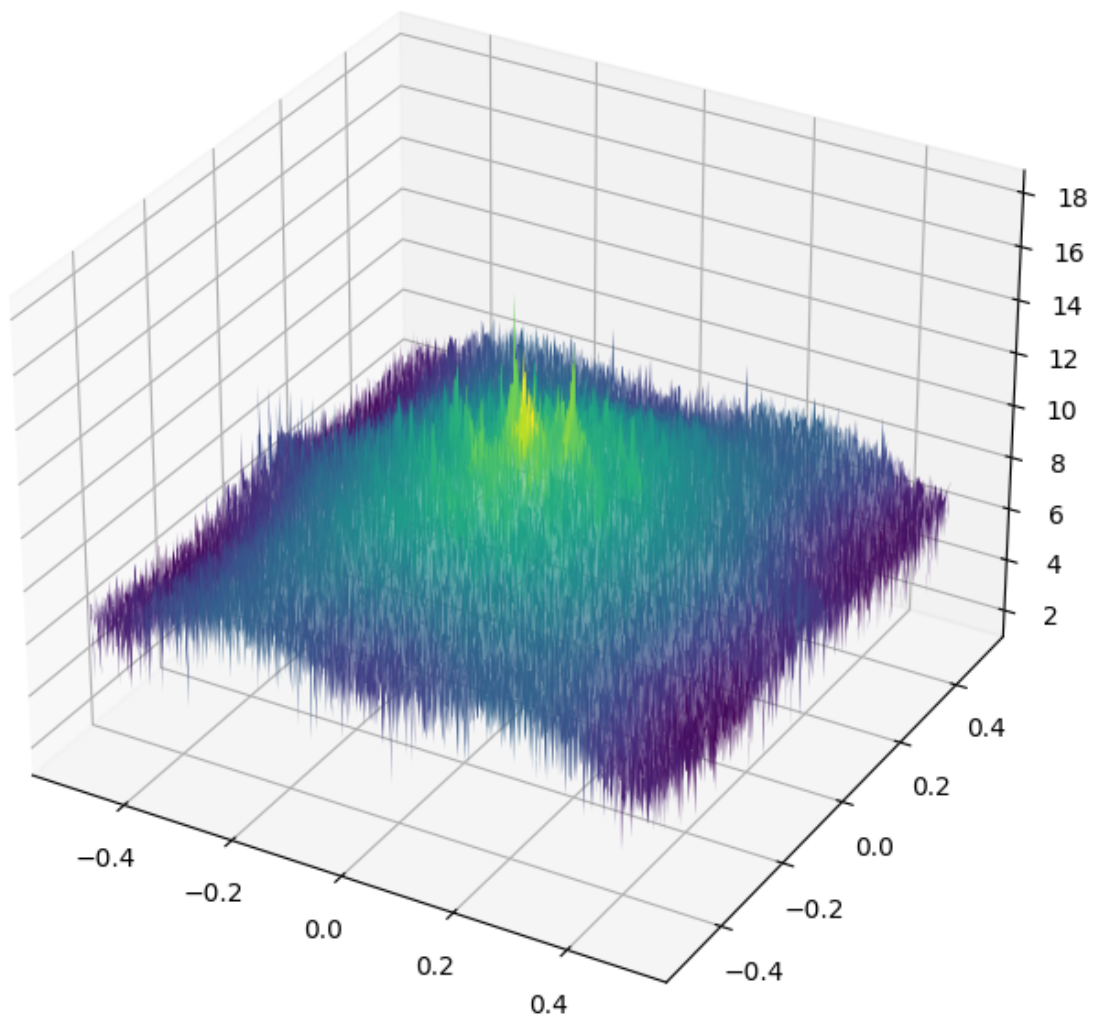
## Espectro 3D - imgLen



## Plotando imagem newspaper\_shot\_woman

```
In [ ]: plotar_espectro_3D(imgNS, "imgNS")
```

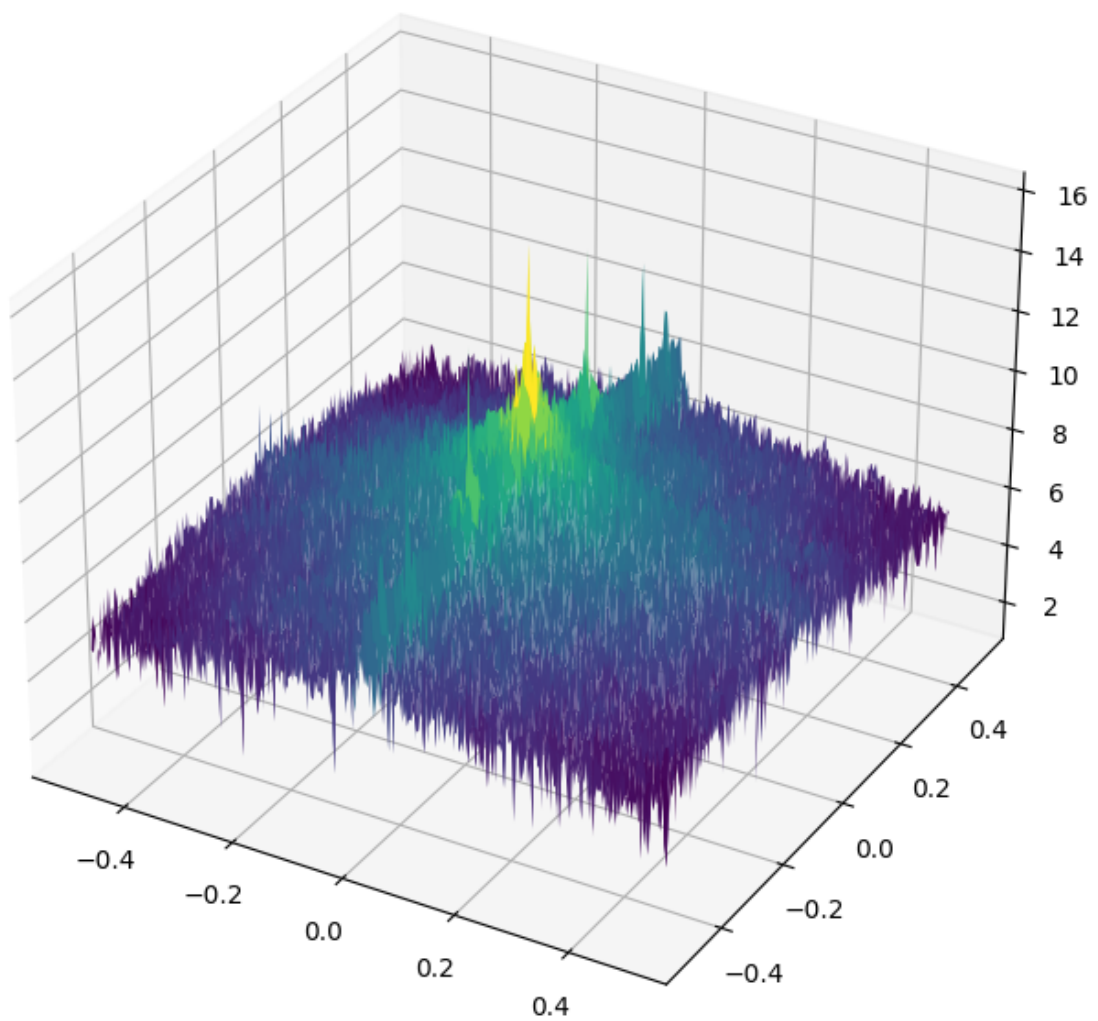
## Espectro 3D - imgNS



## Plotando Imagem periodic\_noise

```
In [ ]: plotar_espectro_3D(imgPeriodic, "imgPeriodic")
```

## Espectro 3D - imgPeriodic



## Plotando Imagem Sinc

```
In [ ]: plotar_espectro_3D(imgSinc,"imgSinc")
```

Espectro 3D - imgSinc

