

Universidade do Estado
de Santa Catarina
UDESC

Filtragem no domínio da
frequência

**Experimento 1: Estudo sobre
transformação de Fourier**

Aluno

Alex Halatiki Vicente

Prof. Responsável

Gilmário

1 de julho de 2023

Introdução

Neste trabalho iremos realizar um estudo experimental sobre filtragem no domínio da frequência.

Fundamentação

A filtragem no domínio da frequência, especificamente a transformação de Fourier, é uma técnica amplamente utilizada no processamento de sinais e imagens. A transformação de Fourier permite representar um sinal no domínio do tempo em termos de suas componentes de frequência. Essa representação no domínio da frequência facilita a análise e manipulação do sinal, especialmente quando se deseja realizar operações como filtragem.

A transformação de Fourier é uma ferramenta matemática que converte um sinal do domínio do tempo para o domínio da frequência. Ela é baseada na decomposição do sinal em uma série de componentes senoidais de diferentes frequências. A transformada de Fourier de um sinal contínuo é chamada de transformada de Fourier contínua, enquanto a transformada de Fourier de um sinal discreto (amostrado) é chamada de transformada de Fourier discreta.

Uma das principais aplicações da transformação de Fourier é na filtragem de sinais. A ideia básica é que diferentes frequências de um sinal podem ser separadas no domínio da frequência. Ao aplicar uma função de filtragem no domínio da frequência, é possível atenuar ou eliminar determinadas componentes de frequência do sinal.

Existem dois tipos principais de filtros no domínio da frequência: filtros passa-baixa e filtros passa-alta. Um filtro passa-baixa permite a passagem de frequências abaixo de um determinado limite, enquanto atenua ou elimina as frequências acima desse limite. Por outro lado, um filtro passa-alta permite a passagem de frequências acima de um determinado limite, atenuando ou eliminando as frequências abaixo desse limite.

A aplicação de um filtro no domínio da frequência envolve os seguintes passos:

Transformação de Fourier: O sinal de entrada é transformado do domínio do tempo para o domínio da frequência utilizando a transformada de Fourier. Isso resulta em um espectro de frequência que representa as diferentes componentes de frequência presentes no sinal.

Filtragem: A função de filtragem é aplicada ao espectro de frequência. Essa função pode ser projetada de acordo com os requisitos específicos do filtro desejado, como a frequência de corte para um filtro passa-baixa. A função de filtragem pode ser representada como uma máscara que multiplica o espectro de frequência.

Transformação inversa de Fourier: O espectro filtrado no domínio da frequência é transformado de volta para o domínio do tempo utilizando a transformada inversa de Fourier. Isso resulta no sinal filtrado, no qual as componentes de frequência indesejadas foram atenuadas ou eliminadas.

Etapa Experimental

Seguindo o tutorial descrito na tarefa para implementar a transformação de Fourier, temos o código abaixo que busca eliminar as linhas verticais da imagem folhas1_Reticulada.jpg.

```
def fourier_masker_ver(image, i):
    f_size = 15
    dark_image_grey_fourier = np.fft.fftshift(np.fft.fft2(image))
    dark_image_grey_fourier[:590, 955:965] = i
    dark_image_grey_fourier[620:, 955:965] = i
    fig, ax = plt.subplots(1,3,figsize=(15,15))
    ax[0].imshow(np.log(abs(dark_image_grey_fourier)), cmap='gray')
    ax[0].set_title('Masked Fourier', fontsize = f_size)
    ax[1].imshow(image, cmap = 'gray')
    ax[1].set_title('Image', fontsize = f_size);
    ax[2].imshow(abs(np.fft.ifft2(dark_image_grey_fourier)),
                  cmap='gray')
    ax[2].set_title('Transformed Image',
                    fontsize = f_size);
    plt.show()

    return abs(np.fft.ifft2(dark_image_grey_fourier))
```

Assim, alterando a função implementada, temos o código abaixo que busca eliminar as linhas horizontais da imagem folhas1_Reticulada.jpg.

```
def fourier_masker_ver(image, i):
    f_size = 15
    dark_image_grey_fourier = np.fft.fftshift(np.fft.fft2(image))
    dark_image_grey_fourier[595:605, :940] = i
    dark_image_grey_fourier[595:605, 985:] = i
    fig, ax = plt.subplots(1,3,figsize=(15,15))
    ax[0].imshow(np.log(abs(dark_image_grey_fourier)), cmap='gray')
    ax[0].set_title('Masked Fourier', fontsize = f_size)
    ax[1].imshow(image, cmap = 'gray')
    ax[1].set_title('Image', fontsize = f_size);
    ax[2].imshow(abs(np.fft.ifft2(dark_image_grey_fourier)),
                  cmap='gray')
    ax[2].set_title('Transformed Image',
                    fontsize = f_size);
    plt.show()

    return abs(np.fft.ifft2(dark_image_grey_fourier))
```

Juntando os procedimentos anteriores, temos o código abaixo que busca eliminar as linhas verticais e horizontais da imagem folhas1_Reticulada.jpg.

```
def fourier_masker_ver(image, i):
    f_size = 15
    dark_image_grey_fourier = np.fft.fftshift(np.fft.fft2(image))
    dark_image_grey_fourier[:590, 955:965] = i
    dark_image_grey_fourier[620:, 955:965] = i
    dark_image_grey_fourier[595:605, :940] = i
    dark_image_grey_fourier[595:605, 985:] = i
    fig, ax = plt.subplots(1,3,figsize=(15,15))
    ax[0].imshow(np.log(abs(dark_image_grey_fourier)), cmap='gray')
    ax[0].set_title('Masked Fourier', fontsize = f_size)
    ax[1].imshow(image, cmap = 'gray')
    ax[1].set_title('Image', fontsize = f_size);
    ax[2].imshow(abs(np.fft.ifft2(dark_image_grey_fourier)),
                  cmap='gray')
    ax[2].set_title('Transformed Image',
                  fontsize = f_size);
    plt.show()

    return abs(np.fft.ifft2(dark_image_grey_fourier))
```

Com Fourier implementada, temos o código abaixo que aplica a transformação na imagem folhas1_Reticulada.jpg e utiliza o teste de similaridade estrutural (SSIM) e o teste de Erro Médio Quadrado (MSE) para determinar o nível de semelhança entre a imagem original e a recuperada.

```
import numpy as np
import matplotlib.pyplot as plt
from skimage.io import imread
from skimage.metrics import structural_similarity as ssim
from skimage.metrics import mean_squared_error

def fourier_masker_ver(image, i):
    f_size = 15
    dark_image_grey_fourier = np.fft.fftshift(np.fft.fft2(image))
    dark_image_grey_fourier[:590, 955:965] = i
    dark_image_grey_fourier[620:, 955:965] = i
    dark_image_grey_fourier[595:605, :940] = i
    dark_image_grey_fourier[595:605, 985:] = i
    fig, ax = plt.subplots(1,3,figsize=(15,15))
    ax[0].imshow(np.log(abs(dark_image_grey_fourier)), cmap='gray')
    ax[0].set_title('Masked Fourier', fontsize = f_size)
    ax[1].imshow(image, cmap = 'gray')
    ax[1].set_title('Image', fontsize = f_size);
    ax[2].imshow(abs(np.fft.ifft2(dark_image_grey_fourier)),
                  cmap='gray')
    ax[2].set_title('Transformed Image',
                  fontsize = f_size);
    plt.show()

    return abs(np.fft.ifft2(dark_image_grey_fourier))

img = imread('folhas1.jpg')

img_grade = imread('folhas1_reticulada.jpg')

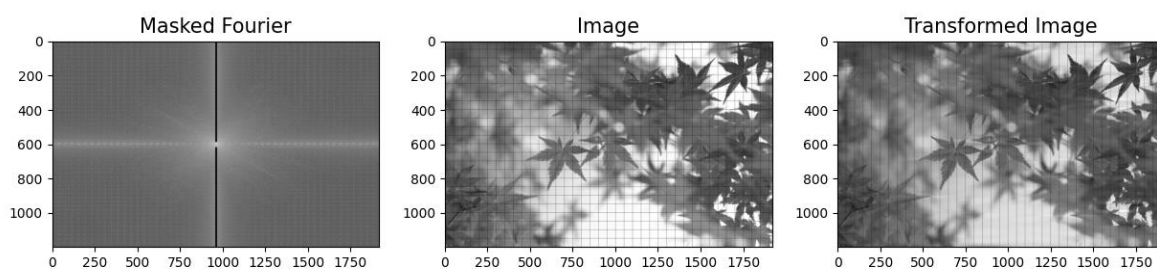
img_fourier = fourier_masker_ver(img_grade, 1)

mse_fourier = mean_squared_error(img, img_fourier)
ssim_fourier = ssim(img, img_fourier,
                    data_range=img_fourier.max() - img_fourier.min())

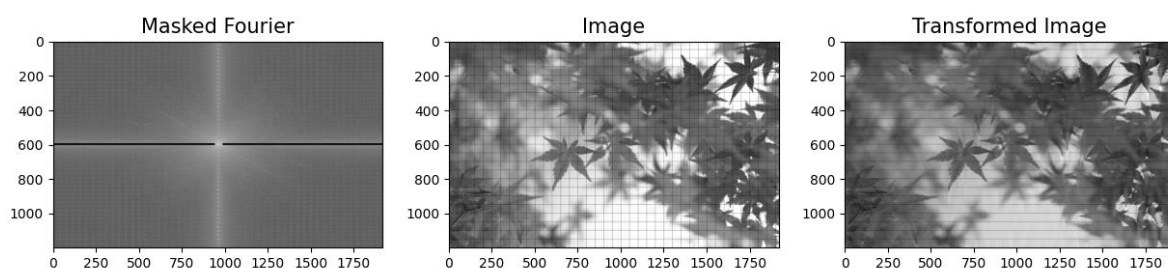
print(f'MSE: {mse_fourier}')
print(f'SSIM {ssim_fourier}')
```

Resultados

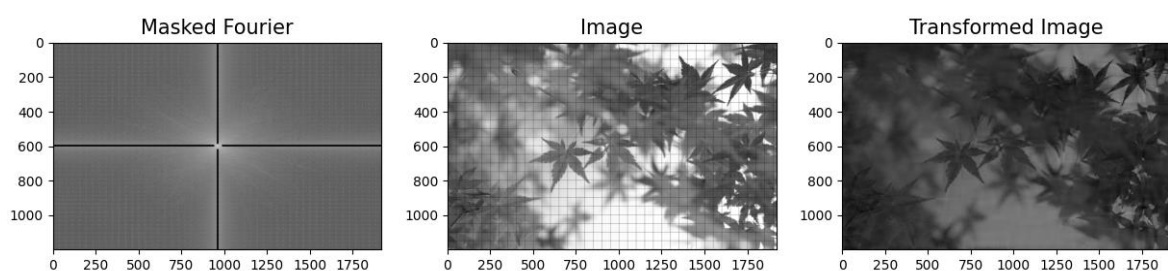
Removendo as linhas horizontais:



Removendo as linhas verticais:



Removendo as linhas horizontais e verticais:



Percebe-se que aplicando o método para retirar as linhas verticais ou horizontais isoladamente cada uma, temos um resultado aparentemente bom. Porém, ao tentar retirar ambas as linhas, tanto verticais como horizontais, é notório que temos um resultado bom, mas percebe-se também, que a imagem acaba escurecendo bastante.

Por fim, temos o resultado do teste de similaridade estrutural (SSIM) e o teste de Erro Médio Quadrado (MSE) aplicados a imagem folhas1_Reticulada.jpg após transformação de Fourier comparada com a imagem original folhas1.jpg:

```
MSE: 177.58899230373558  
SSIM 0.9292420471937237
```

O SSIM é uma medida de similaridade estrutural entre duas imagens. Ele compara a estrutura local, o contraste e a luminância das imagens para determinar o quão semelhantes elas são. O valor do SSIM varia de 0 a 1, sendo 1 o valor máximo de similaridade entre as imagens. Quanto mais próximo de 1 for o valor do SSIM, maior será a similaridade estrutural entre as imagens. O valor de 0.9292420471937237, indica uma boa similaridade entre as imagens.

Por outro lado, o MSE é uma medida do erro médio quadrado entre as imagens. Ele calcula a média dos quadrados das diferenças pixel a pixel entre a imagem original e a imagem processada ou comprimida. Quanto menor o valor do MSE, menor é o erro médio entre as imagens. O valor de 177.58899230373558, indica que há um nível de erro médio relativamente alto entre as imagens.

Com isso, conseguimos perceber que foi possível recuperar bastante da imagem original, eliminando as linhas indesejadas. Porém, o procedimento executado escureceu a imagem em comparação com a original.

Conclusões

Concluindo, a filtragem no domínio da frequência, por meio da transformação de Fourier, é uma técnica poderosa e amplamente aplicada no processamento de sinais e imagens. Essa abordagem permite a separação e manipulação das componentes de frequência de um sinal, possibilitando a aplicação de filtros para atenuar ou eliminar frequências indesejadas. Embora apresente vantagens, como flexibilidade e eficiência computacional, é importante considerar suas limitações, como a necessidade de um sinal estacionário e a possibilidade de introdução de artefatos. Apesar disso, a filtragem no domínio da frequência continua desempenhando um papel fundamental em diversas áreas, impulsionando avanços tecnológicos e aprimorando a qualidade dos sinais processados.