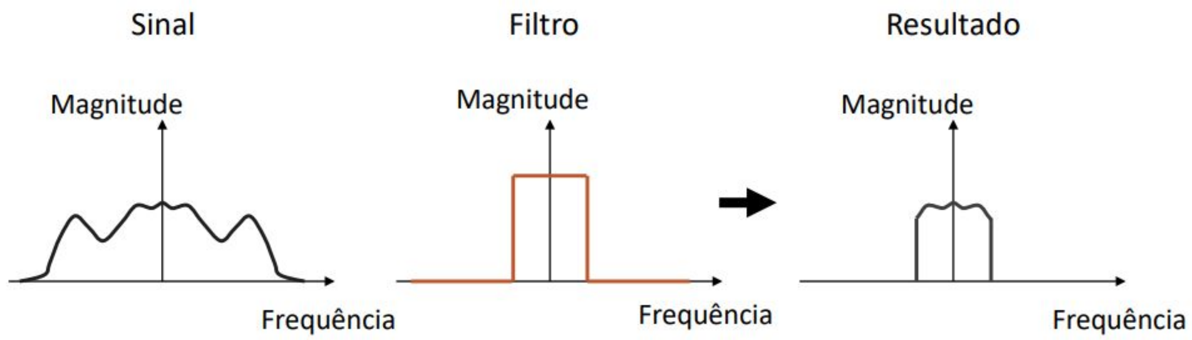
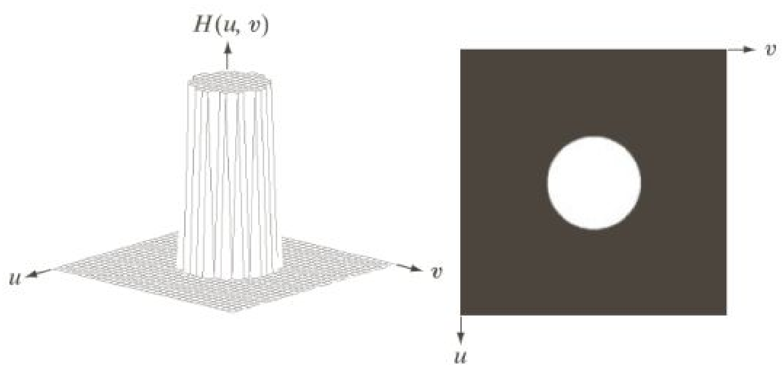
Trabalho 12- Tópicos em Informática 11
Tema 2: Filtro passa baixa e passa-alta butterworth
Membros:
Guilherme Nishi Kanashiro, RA: 628298
Leonardo Utida Alcântara, RA: 628182
Rodolfo Krambeck Asbahr, RA: 628042
Tiago Bachiega de Almeida, RA: 628247

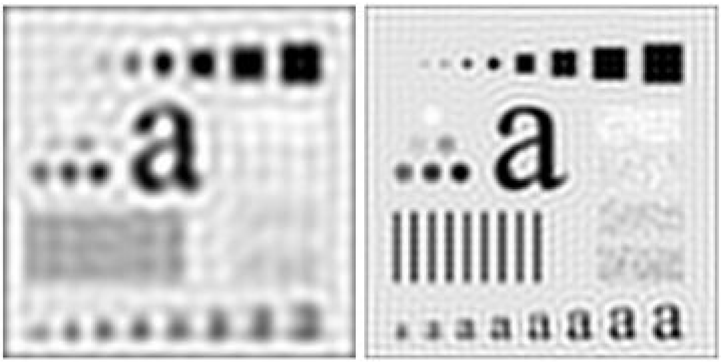
Os filtros Butterworth foram descritos pela primeira vez pelo engenheiro britânico S. Butterworth, em sua publicação "On the Theory of Filter Amplifiers". Esse tipo de filtro é caracterizado por ter uma faixa de passagem e de rejeição planas e uma região de transição moderada.

A ideia de um filtro passa-alta ou passa-baixa do estilo butterworth é filtrar uma determinada região (nesses casos de valores máximos ou mínimos) de frequências, porém ao contrário dos filtros passa-alta e passa-baixa ideais, a transição entre a faixa de passagem e a faixa de rejeição não é feita de forma abrupta. As figuras abaixo mostram casos de filtros passa baixa com esse corte de frequências abrupto.





Um maior controle da suavidade dessa região de transição é necessário pois em certos casos um corte de frequências abrupto pode gerar efeitos indesejados na filtragem na imagem, como por exemplo o efeito de ondas. As imagens abaixo mostram figuras filtradas com um filtro passa-baixa de corte em uma imagem 2D, onde ocorre um efeito chamado “ringing” que é criado por esse corte abrupto das frequências altas (que definem as bordas da imagem).



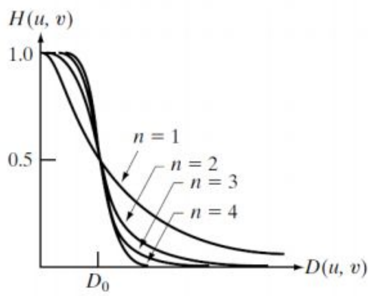
Esse problema pode ser solucionado a partir do uso de um filtro que tenha uma transição mais suave entre as frequências que são bloqueadas e as que são permitidas. Podemos utilizar um filtro gaussiano no domínio da frequência ou então um filtro do estilo passa-faixa do estilo butterworth.

O filtro butterworth possui dois parâmetros ***n*** e $\boldsymbol{D_0}$. $\boldsymbol{D_0}$ define onde irá ocorrer o corte de frequências. Já o valor ***n*** define a suavidade da transição entre a faixa de rejeição e aceitação, quanto maior o valor de ***n*** mais abrupta é a transição. Desse modo, se optarmos por um valor de ***n*** muito alto, teremos um efeito muito similar aos filtros passa-alta ou passa-baixa convencionais. A função que define o filtro, juntamente com uma representação visual do filtro na versão 1D pode ser vista abaixo:

Passa-baixa

$$H(\mu,\nu) = \frac{1}{1+\left(\frac{D(\mu,\nu)}{D_0}\right)^{2n}}$$

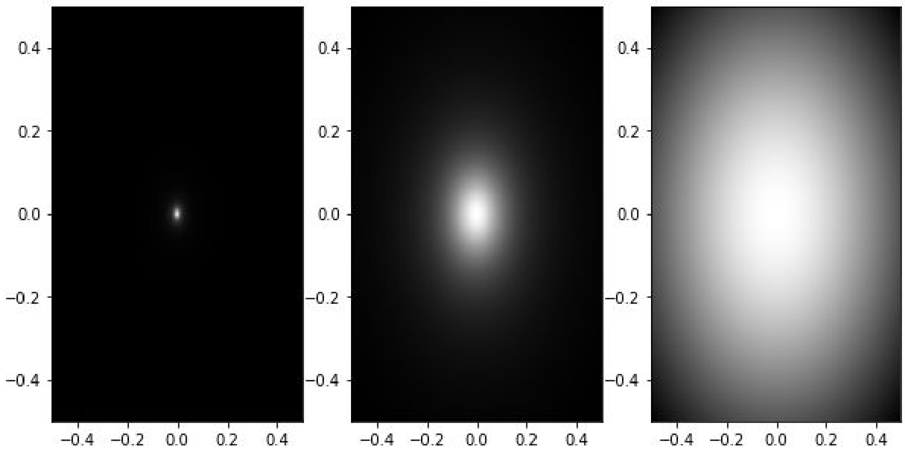
$$D(\mu,\nu) = \sqrt{\mu^2+\nu^2}$$



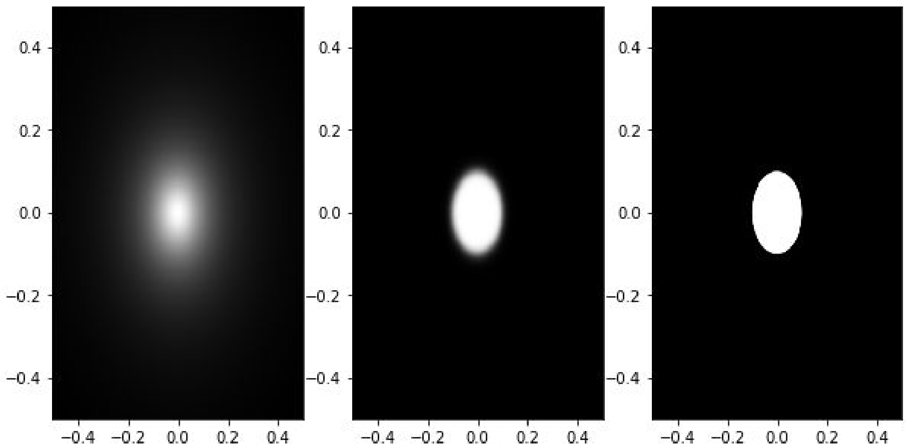
Passa-alta

$$H(\mu,\nu) = \frac{1}{1+\left(\frac{D_0}{D(\mu,\nu)}\right)^{2n}}$$

O filtro também pode ser criado para o caso 2D, para isso a função é aplicada para cada combinação de índices de pixels da imagem, só que no domínio da frequência. Abaixo estão exemplos de filtros butterworth 2d. Na primeira imagem foi mantido um valor fixo para ***n*** e os valores de $\boldsymbol{D_0}$ foram variados. Já na segunda imagem o foi feito o contrário.



Valores de ***n*** fixo e ***D_0*** variável.



Valores de ***n*** variável e ***D_0*** fixo.

Nas imagens abaixo resultados da filtragem passa baixa usando o filtro de Butterworth implementado, pode-se perceber que usando um valor de n muito alto o efeito de ringing, citado anteriormente, se torna visível. Na esquerda, foi usado o valor $\mathbf{D_0}$ = 0.1 e o valor de **n**=1 . E na direita, foi usado o valor $\mathbf{D_0}$=0.1 e **n**=80.

Quanto maior o valor de **n**, mais perto de ser um filtro passa alto/passa baixo comum. Quando se deseja cortar uma faixa específica de frequência, o ideal é usar o filtro de Butterworth com um valor de **n** alto. Porém em situações onde se deseja um corte mais suave o ideal é procurar usar valores de n mais baixos, por exemplo usar o filtro em uma imagem que tem mais detalhes e bordas.

No código abaixo, temos a função do filtro de Butterworth que recebe como parâmetro uma imagem, o valor de $\mathbf{D_0}$, o valor de **n** e o **tipo do filtro** (passa alto ou passa baixo). E retorna o filtro de Butterworth segunda a equação descrita anteriormente para o tipo de filtro e os valores de $\mathbf{D_0}$ e **n**. Para calcular cada elemento do filtro, foram utilizados os arrays freq_r e freq_c que contém o valor das frequências das amostras da transformada discreta de Fourier, na forma [-fmax,...,0,...,fmax], calculado com base no tamanho da imagem.

```
def filtro_butterworth(img, D_0, n, filter_type):
    num_rows, num_cols = img.shape
    freq_r = fftfreq(num_rows)
    freq_c = fftfreq(num_cols)
    freq_r = fftshift(freq_r)
    freq_c = fftshift(freq_c)

    butter_filter = np.zeros([num_rows,num_cols])
    for row in range(num_rows):
        for col in range(num_cols):
            D = np.sqrt(freq_r[row]**2+freq_c[col]**2)
            if filter_type == "low":
                butter_filter[row,col] = (1/(1+((D/D_0)**(2*n))))
            if filter_type == "high":
                if D == 0:
                    D = 0.0001
                butter_filter[row,col] = (1/(1+((D_0/D)**(2*n))))

    return butter_filter
```

Como o filtro está no domínio da frequência é preciso transformar a imagem também. Isso é mostrado no código abaixo, onde a imagem filtrada, Fimg_filtered é a multiplicação (convolução) de cada elemento da imagem pelo correspondente do filtro, lp_filter1.

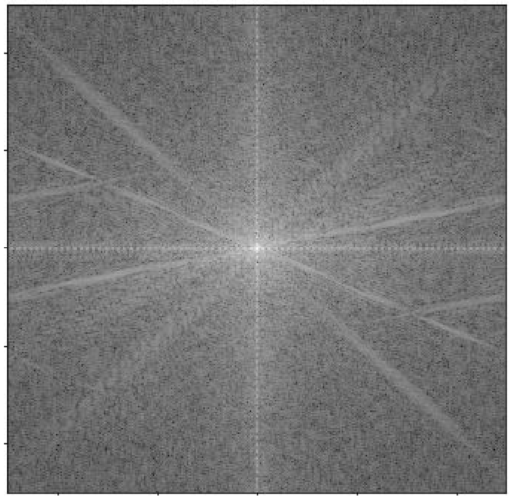
```
Fimg = fft2(img_padded)
Fimg = fftshift(Fimg)
lp_filter1 = filtro_butterworth(img_padded, 0.01, 1, "low")
Fimg_filtered1 = Fimg * lp_filter1
```

Como os filtros são usados no domínio da frequência é preciso fazer a transformada inversa de Fourier após a aplicação do filtro para mostrar a imagem. Abaixo, temos um exemplo do procedimento feito para realizar a transformada inversa.

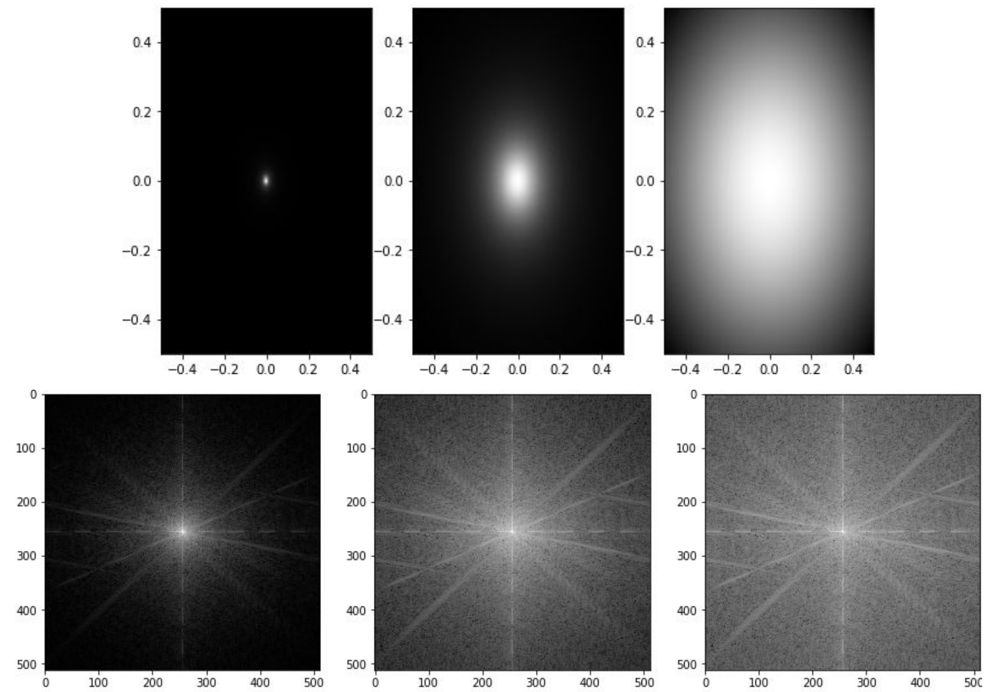
```
Fimg_filtered1 = fftshift(Fimg_filtered1)
img_filtered1 = np.real(ifft2(Fimg_filtered1))
img_filtered_final1 = img_filtered1[0:num_rows, 0:num_cols]
```

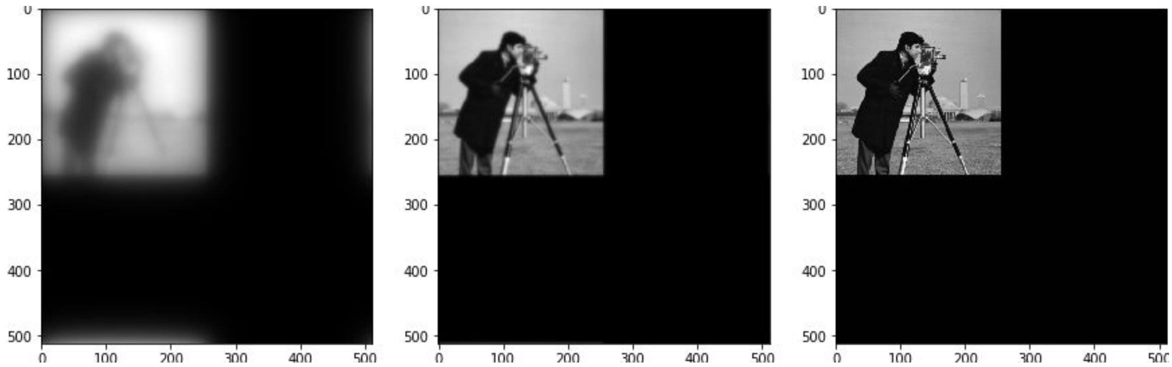
Exemplos do funcionamento do código apresentados no notebook:

Para testarmos o funcionamento da função descrita acima foram feitos diversos testes variando os valores de **n** e $\mathbf{D_0}$ para entender melhor como cada um influencia no resultado final. Realizamos os testes com a imagem do cameraman que usamos durante as aulas. Abaixo estão a imagem (à esquerda) e a transformada de fourier da imagem (à direita), já com uma transformação linear logarítmica para melhorar o contraste. Em cada teste foram criados 3 filtros com diferentes propriedades variando apenas uma das variáveis para identificar como ela altera o resultado final. A primeira linha de imagens corresponde aos 3 filtros butterworth criados em cada teste, já a segunda linha de imagens corresponde à multiplicação da transformada de Fourier da imagem com os filtros e por fim a última linha de imagens corresponde à transformada inversa dessa multiplicação, retornando a imagem filtrada.



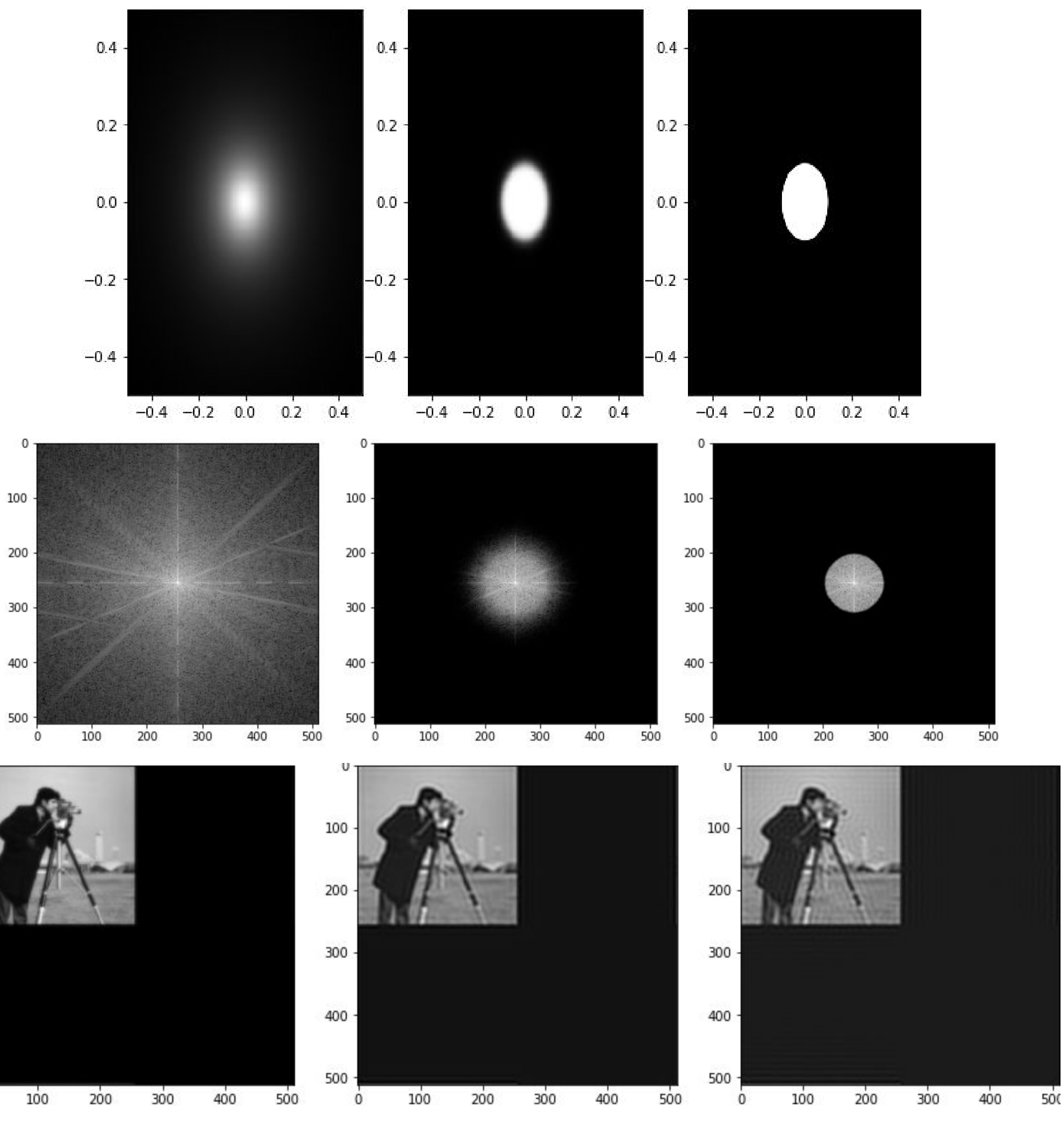
Teste 1: Filtro passa-baixa fixando os valores de **n** em 1 e variando os valores de $\mathbf{D_0}$ (0,01, 0.1 e 0.7).





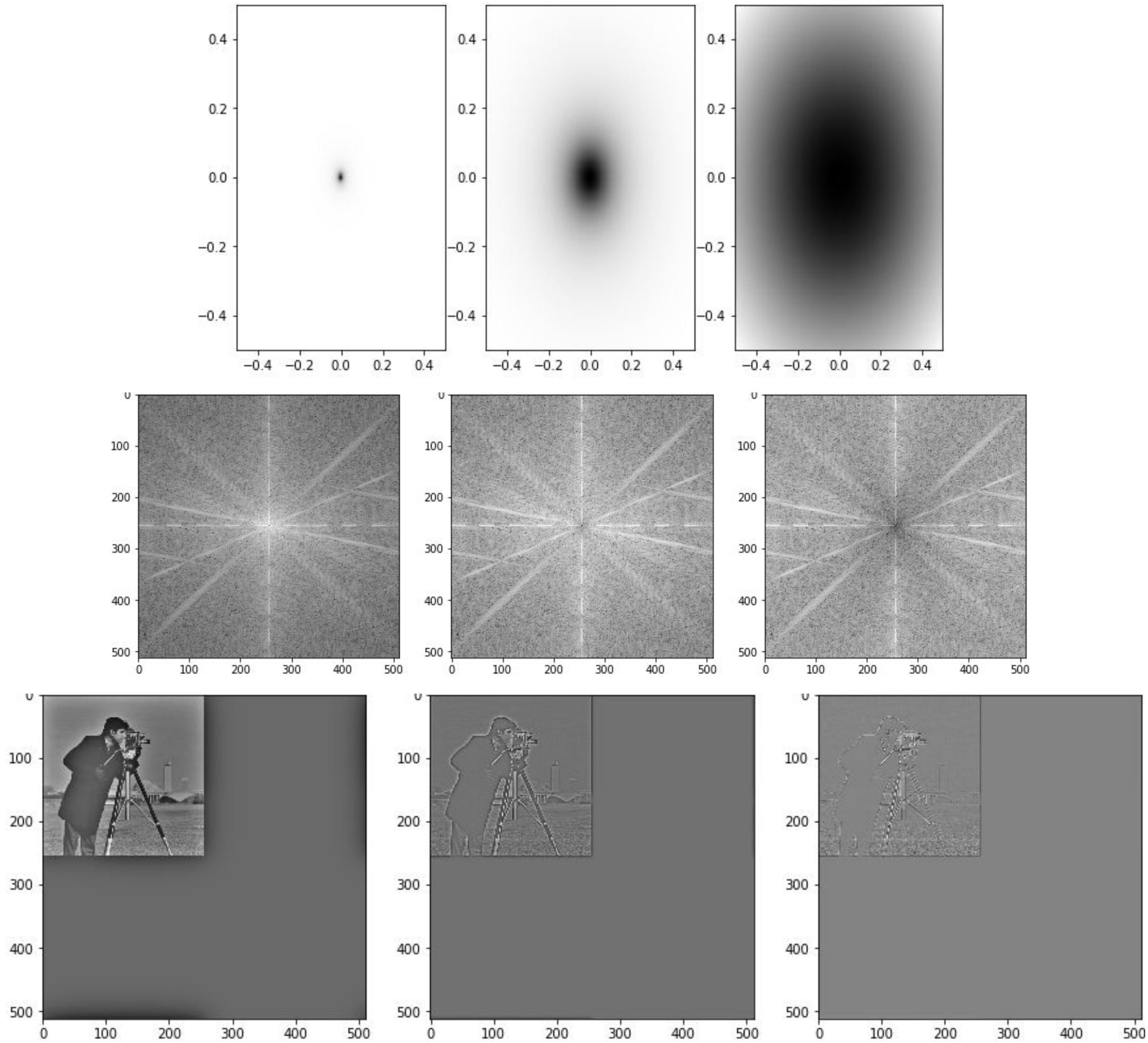
Como o filtro criado era um passa-baixa, o valor de $\mathbf{D_0}$ define onde o corte será feito. Como nos nossos testes o valor de **n** ficou fixo em 1 a transição é mais suave, desse modo criando uma suavização mais gradual na imagem. Porém é possível notar que para um $\mathbf{D_0}$ mais baixo, as frequências altas foram eliminadas, desse modo eliminando as bordas e suavizando mais a imagem. Já aumentando o valor de $\mathbf{D_0}$ a imagem não teve tanta suavização.

Teste 2: Filtro passa-baixa fixando os valores de $\mathbf{D_0}$ em 0.1 e variando os valores de **n** (1, 7 e 80).



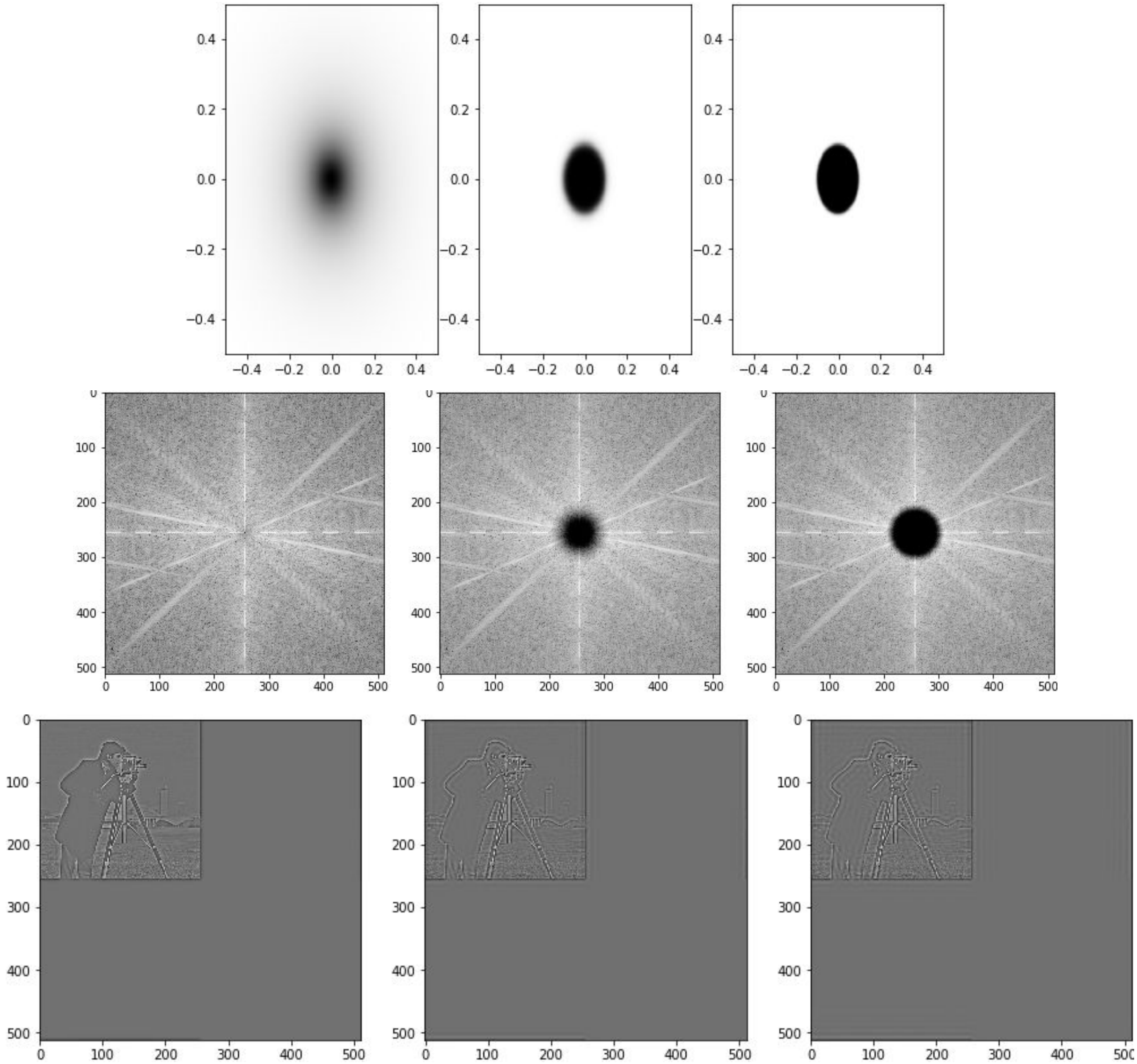
Nesse teste é possível ver diferentes níveis de suavidade na transição do filtro. Na última imagem o valor de **n** é muito alto, dessa forma o filtro se comporta igual um filtro passa-baixa comum e por isso é possível identificar o efeito de “ringing”. Vemos que com um valor de **n** mais baixo a suavização ocorre de maneira mais “leve”, porém não causa esse tipo de efeito na imagem.

Teste 3: Filtro passa-alta fixando os valores de **n** em 1 e variando os valores de $\mathbf{D_0}$ (0,01, 0.1 e 0.7).



Nesse teste vemos que $\mathbf{D_0}$ mais uma vez define onde será o corte do nosso filtro passa-alta. Desse modo para um valor baixo de $\mathbf{D_0}$ as bordas da imagem não ficaram tão evidentes, porém mais valores mais altos as regiões mais planas da imagem foram filtradas deixando apenas as bordas da imagem.

Teste 4: Filtro passa-baixa fixando os valores de $\mathbf{D_0}$ em 0.1 e variando os valores de **n** (1, 7 e 80).



Nesse último teste temos diferentes níveis de suavidade para a transição do filtro passa-alta. Podemos ver que no filtro mais a direito o valor de **n** é bem alto desse modo, o corte nas frequências é bem abrupto de maneira similar a um filtro passa-alta convencional. Isso causa algumas manchas na imagem em locais onde não tem bordas e podemos ver novamente o efeito de “ringing”, principalmente no topo da imagem. Vemos que com um valor de **n** mais baixo, esse tipo de efeito some e as bordas ficam mais nítidas (imagem do meio).