

Dados do relatório Lista 4: filtros no domínio da Frequência

Matéria: Processamento Digital de Imagens (PDI) **código:** CMP1084-A01

Aluno: Vitor de Almeida Silva

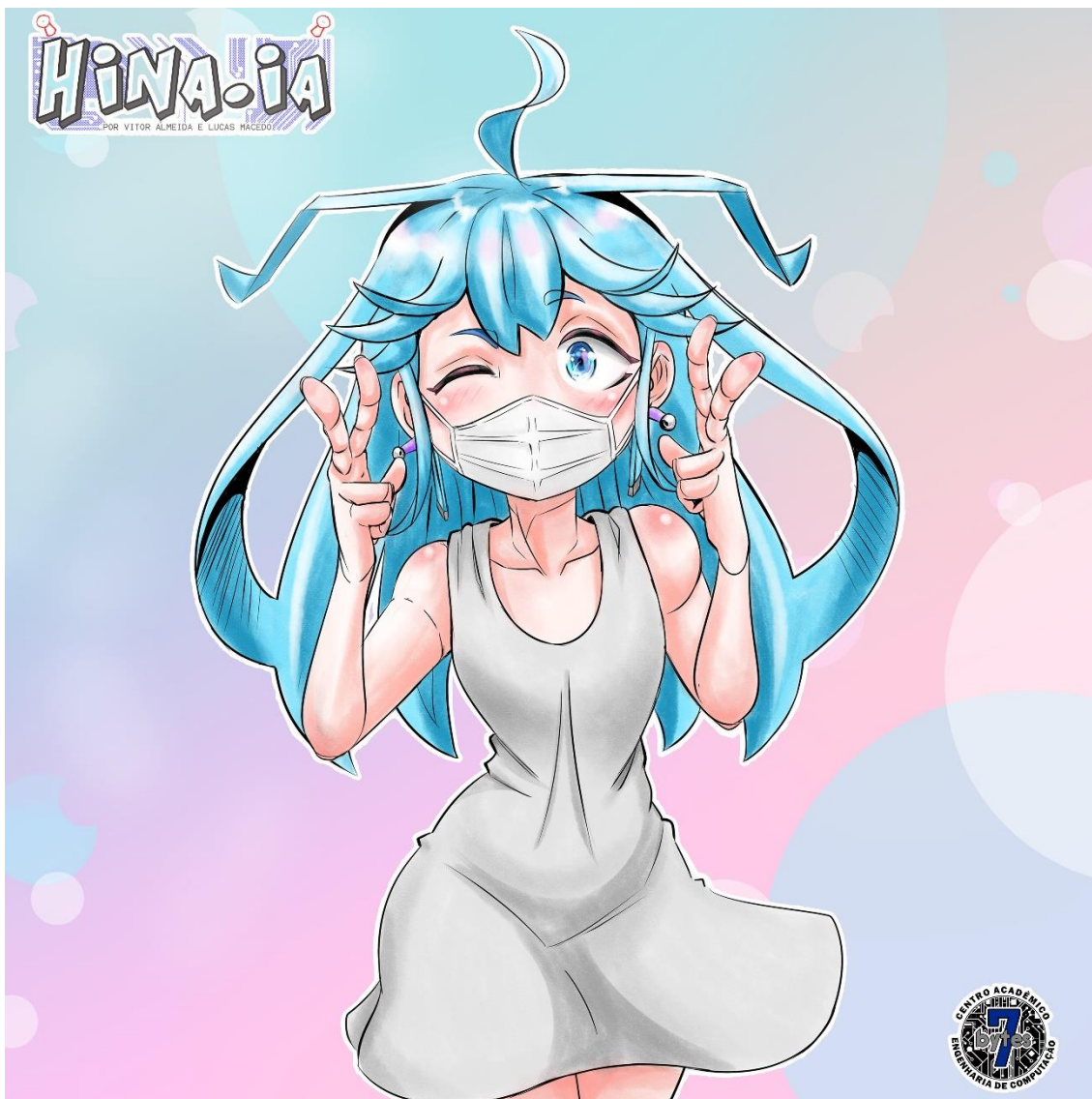
Matrícula: 20161003305497

ENUNCIADO

1. Carregue a imagem escolhida e transforme em escala de cinza e execute a transformada e mostre sua magnitude.
2. Carregue a imagem transforme em escala de cinza e execute a transformada, e em seguida crie uma máscara circular e aplique direto na magnitude da imagem escolhida para que seja executada a **passa-baixa ideal**. Volte a imagem para o domínio do tempo e mostre o resultado.
3. Carregue a imagem transforme em escala de cinza e execute a transformada, e em seguida crie a máscara *Butterworth* ($D_0 = 180$ e $N=2$) e aplique direto na magnitude da imagem escolhida. Volte a imagem para o domínio do tempo e mostre o resultado.
4. Carregue a imagem transforme em escala de cinza e execute a transformada, e em seguida crie uma máscara circular e aplique direto na magnitude da imagem escolhida para que seja executada a **passa-alta ideal**. Volte a imagem para o domínio do tempo e mostre e **comente** o resultado.
5. Utilize a imagem resultante na questão 4 e refaça-a questão com ela, explique se teve alguma diferença, e mostre a magnitude antes de aplicar a máscara de novo. Volte a imagem para o domínio do tempo e mostre o resultado.
6. Refaça questão 2 com uma máscara retangular, dimensões a sua escolha. Explique a diferença visual com a imagem resultante da questão 2.
7. Execute a questão 2 mas a imagem resultante deve ser colorida.

A imagem selecionada para realização da presente lista, é mostrada na Figura 1.

Figura 1: img1



Fonte: Mangá Hina.IA (Autorial) - < <https://drive.google.com/drive/folders/14E7y-Gz5n1wG28FOtepOEwYoiRD3E26K> >

1. Carregue a imagem escolhida e transforme em escala de cinza e execute a transformada e mostre sua magnitude.

Resposta:

A transformada de Fourier tem a propriedade de transformar uma imagem no domínio do tempo para o domínio da frequência, por meio de uma varredura da imagem produzida por um somatório, este, é mostrado na Figura 2 (GONZALEZ, WOODS, 2010).

Figura 2: Transformada discreta de Fourier

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

Fonte: (GONZALEZ, WOODS, 2010).

Com isso, a imagem que é, originalmente, formada por pixels com valores de intensidades, passa para um formato onde cada píxel (xf,yf), representa os valores da frequência de determinada intensidade em toda a imagem original (GONZALEZ, WOODS, 2010).

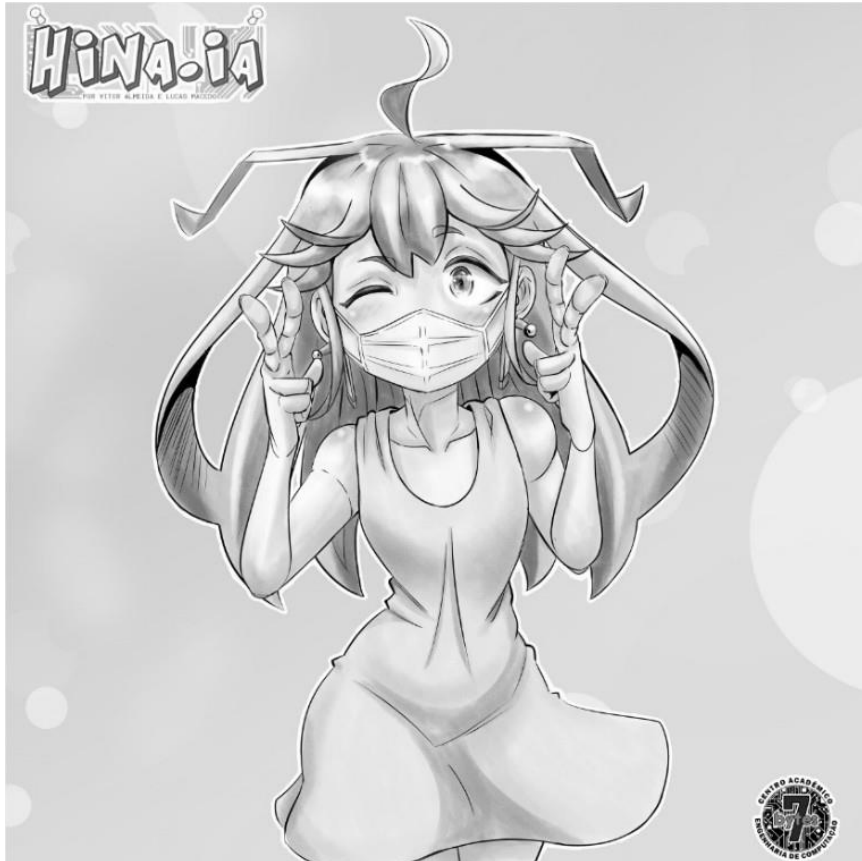
Visualizar uma imagem no espectro da frequência não faz muito sentido, já que, é impossível fazer associações diretas de F(u,v) com f(x,y). porém, é possível se retirar informações úteis por meio da análise da magnitude dos pixels, como por exemplo:

- Cada termo de F(u,v) contém todos os valores de f(x,y) modificados pelos valores dos termos exponenciais;
- É possível associar frequências na imagem transformada com padrões de intensidade na imagem original;
- **Baixar frequências** correspondem aos componentes de intensidade de variação lenta em uma imagem (corresponde a variações suaves na intensidade como cores contínua).
- **Altas frequências** correspondem a variações de intensidade cada vez mais rápidas na imagem (bordas de objetos e outros componentes na imagem, que se caracterizam por mudanças abruptas de intensidade).

Com isso, tornar possível a representação visual dos dados de frequência, pode ser feita com o uso de um operação logarítmica ("**log()**") sobre os valores

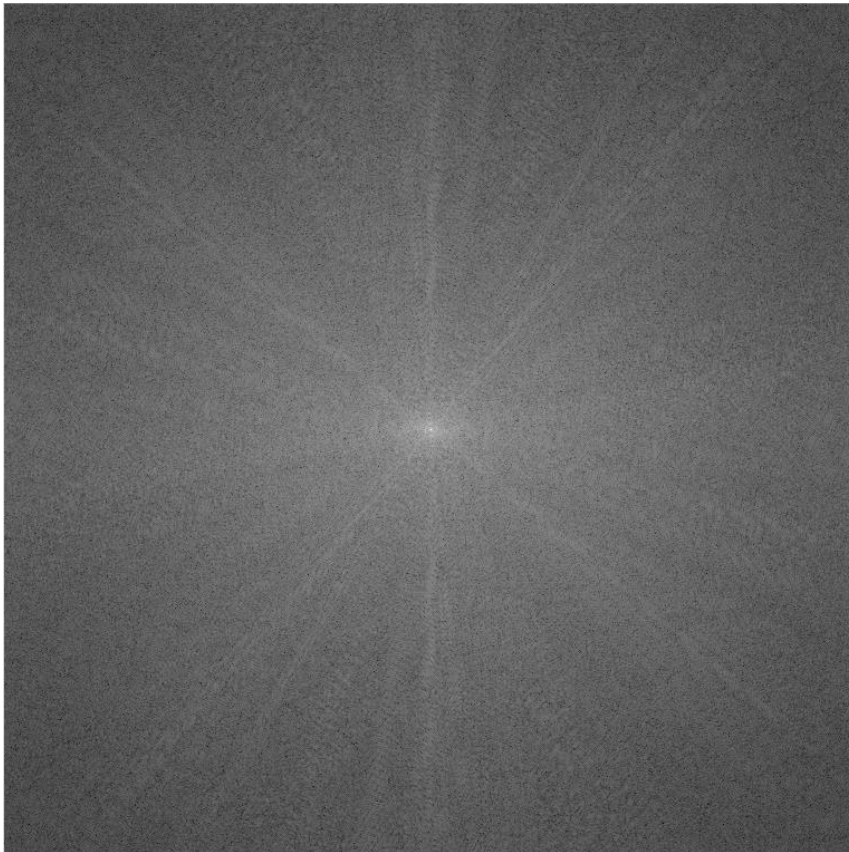
absolutos (“**abs()**”) dos termos da imagem transformada e sua normalização Gonzalez (2010). Também cabe destacar que, a transformada retorna uma matriz com os maiores valores de intensidade nas bordas, para realização de filtragem, deve-se centralizar estes valores por meio de uma função de “**fftShift()**”. As Figuras 3 e 4, mostram a imagem original em escala de cinza e a representação da imagem em magnitude.

Figura 3: imagem em tons de cinza



Fonte: Autoral

Figura 4: Magnitude da imagem



Fonte: Autoral

Código Questão 1:

```
%leitura da imagem
img1=imread('img1.jpg');
imshow(img1(:,:));
title('imagem original');
figure();

imgGray = rgb2gray(img1);

imshow(imgGray(:,:));
title('imagem em escala de cinza');
figure();

imgfft = fft2(imgGray);

imgfftShift = fftshift(imgfft);

imshow(log(abs(imgfftShift)),[]);
title('magnitude da imagem');
figure();
```

2. Carregue a imagem transforme em escala de cinza e execute a transformada, e em seguida crie uma máscara circular e aplique direto na magnitude da imagem escolhida para que seja executada a passa-baixa ideal. Volte a imagem para o domínio do tempo e mostre o resultado pág 176 Gonzales.

Resposta:

Um filtro **passa-baixa ideal**, deixa passar, sem atenuação, todas as frequências em um círculo de raio **D0** a partir da origem e “recorta” (atenua), todas as frequências fora desse círculo. Este diâmetro chamado de D0, é um limiar estabelecido para construção do corte das frequências (D0 seria a **frequência de corte** do filtro). Deste modo, a Figura 5 mostra a função de intervalos que define o filtro passa-baixa ideal circular (GONZALEZ, WOODS, 2010).

Figura 5: Filtro passa-baixa ideal circular

$$H(u,v) = \begin{cases} 1 & \text{se } D(u,v) \leq D_0 \\ 0 & \text{se } D(u,v) > D_0 \end{cases}$$

Fonte: (GONZALEZ, WOODS, 2010).

Onde:

- **D0**: constante positiva, representa o diâmetro ou frequência de corte do filtro;
- **D(u,v)**: é a distância entre um ponto (u,v) e o centro do retângulo de frequência, que é dado pela seguinte equação Eq1:
 - **$D(u,v) = [(u - P/2)^2 + (v - Q/2)^2]^{(1/2)}$** .Eq1
 - **P** e **Q** são as dimensões da máscara (PxQ) (caso fosse uma operação de convolução estes tamanhos deveriam ser $P \geq 2M-1$ e $Q \geq 2N-1$).

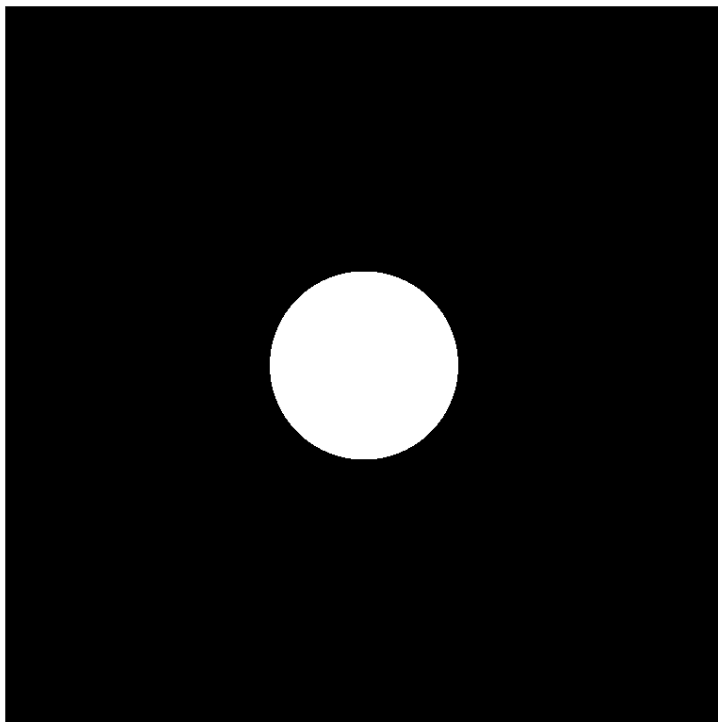
Com isso, após se ter configurado o filtro, é necessário realizar uma transformada rápida de Fourier 2D (**fft2()**) na imagem. Tal transformada, vai passar os valores de intensidade de pixels para valores de magnitude de frequência, que, por sua vez, são valores complexos e reais. Dado isso, basta

realizar uma multiplicação da máscara com a imagem em magnitude de frequência, isso realiza a filtragem, logo, o corte das frequências acima de D_0 .

Para retornar a imagem para o domínio do tempo, é necessário realizar uma inversa da transformada rápida de Fourier 2D (**ifft2()**). Deste modo, como dito anteriormente, quando se realiza a transformada na imagem, valores de intensidades de pixels passam a ser valores de números complexos reais, portanto, é preciso se retirar o valor absoluto (**abs()**) e inteiro deles (**uint8()**).

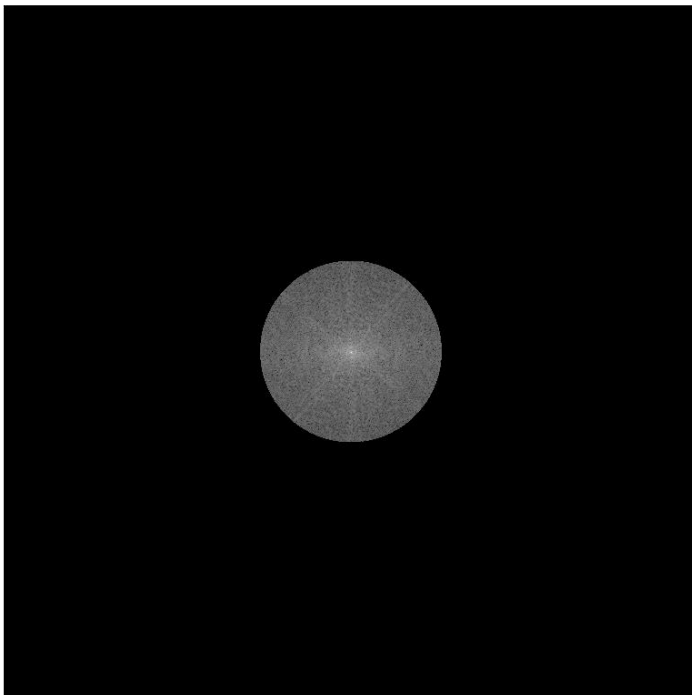
Dado o que foi dito anteriormente, as Figura 6, 7 e 8 mostram a máscara definida com $D_0=100$, a máscara direto na magnitude e a imagem filtrada com filtro passa baixa ideal respectivamente.

Figura 6: Mascara passa baixa ideal circular $D_0=100$



Fonte: Autoral

Figura 7: Mascara direto na magnitude $D_0=100$



Fonte: Autoral

Figura 6: imagem filtrada com filtro passa-baixa ideal circular $D_0=100$



Fonte: Autoral

Por meio da análise da Figura 6, é possível notar que através da filtragem passa-baixa ideal circular, foram obtidos erros do tipo ***ringing*** (texturas semelhantes a riscos ou traços desbotados na imagem em volta das bordas). Está é uma característica deste filtro e, conforme o Do da máscara é aumentado, esses ***ringing*** tendem a se tornar menos aparentes.

Código questão 2:

```
%leitura da imagem
img1=imread('img1.jpg');
imshow(img1(:,:,:));
title('imagem original');
figure();

[x,y,z] = size(img1);

imgGray = rgb2gray(img1);

imshow(imgGray(:,:,:));
title('imagem em escala de cinza');
figure();

imgfft = fft2(imgGray);

imgfftShift = fftshift(imgfft);

imshow(log(abs(imgfftShift)),[]);
title('magnitudo da imagem');
figure();

%gerando mascara de filtro circular ideal passa baixa
Do=100; %diametro do circulo
masck = ideal_low_Pass_Circular_masck(img1,Do);

imshow(masck(:,:,:).*255);
title("mascara filtro passa-baixa ideal circular");
figure();

%realizando filtragem da imagem em magnitude de frequência
imgFiltred = imgfftShift .* masck;

imshow(log(abs(imgFiltred)),[]);
title('maginitude da imagem filtrada');
figure();

%voltando imagem para o domínio do tempo
imgFiltred_temp=uint8( abs ( ifft2(fftshift(imgFiltred)) ) );

imshow(imgFiltred_temp);
title('imagem filtrada ideal low pass Do=' + string(Do));
figure();

%gerando mascara de filtro circular ideal passa baixa
%essa função foi baseada nas equações expostas em
%(GONZALEZ, WOODS, 2010)pág 176
function masck = ideal_low_Pass_Circular_masck (img1,Do)

[x,y,z] = size(img1);

masck = zeros(x,y,'double');
% Do é o diâmentro do circulo, freq de corte
for i=1: x
    for j=1:y
        if( (( i- x/2)^2 + (j - y/2)^2 )^(1/2)) <=Do)
            masck(i,j,:)=1;
        end
    end
end
end
end
```

3. Carregue a imagem transforme em escala de cinza e execute a transformada, e em seguida crie a máscara *Butterworth* ($D_0= 180$ e $N=2$) e aplique direto na magnitude da imagem escolhida. Volte a imagem para o domínio do tempo e mostre o resultado.

Resposta:

Diferentemente do filtro passa-baixa circular ideal, a função de transferência do *butterworth low pass filter* (BLPF) não tem uma descontinuidade abrupta que resulta em um corte perfeito entre frequências passadas e filtradas. Desta forma, a função de transferência do BLPF é mostrada na Figura 7 (GONZALEZ, WOODS, 2010).

Figura 7: *butterworth low pass filter* (BLPF)

$$H(u,v) = \frac{1}{1 + [D(u,v) / D_0]^{2n}}$$

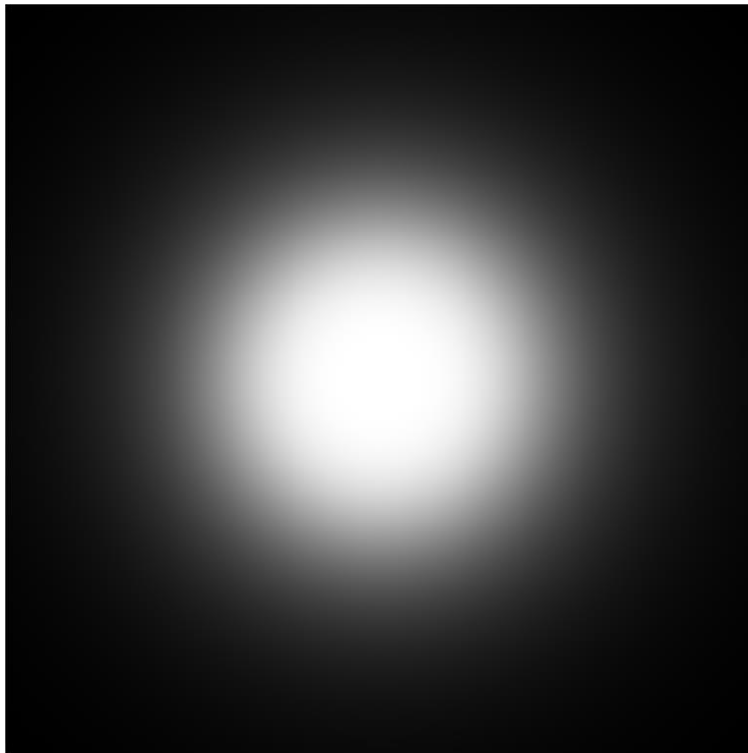
Fonte: (GONZALEZ, WOODS, 2010)

Onde:

- N: é a ordem do filtro;
- D_0 : é a frequência de corte a uma distância D_0 da origem;
- $D(u,v)$: distancia entre o ponto (u,v) e a origem;
- Reduz-se 50% em $\max(D(u,v))$ quando $D(u,v) = D_0$.

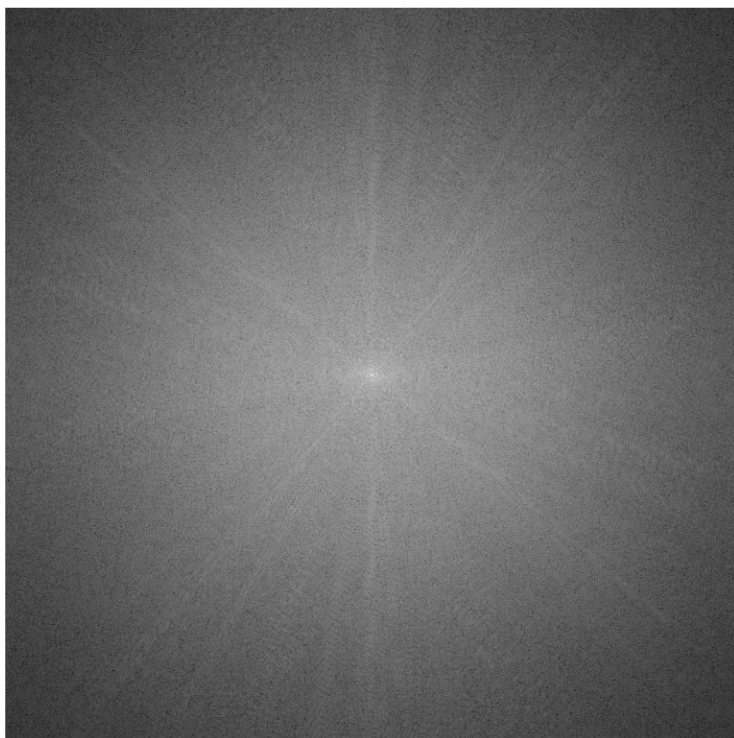
Para a criação de máscaras de transferências suaves, geralmente é aplicado um *locus*. Tal locus, consiste de se reduzir uma determinada fração de seu valor máximo 1 (aqui foi selecionado 50%), em pontos onde $D(u,v) = D_0$. Com esse raciocínio, as Figuras 8, 9 e 10 mostram a máscara do buttherworth, a mascara aplicada direto na magnitude da imagem e a imagem filtrada no domínio do tempo respectivamente com $D_0=180$ e $n=2$.

Figura 8: máscara do filtro butterworth com $n=2$ e $D_0=180$



Fonte: Autoral

Figura 9: Mascara aplicada direto na magnitude com $n=2$ e $D_0=180$



Fonte: Autoral

Figura 10: imagem filtrada com *butterworth low pass* com $n=2$ e $D_0=180$

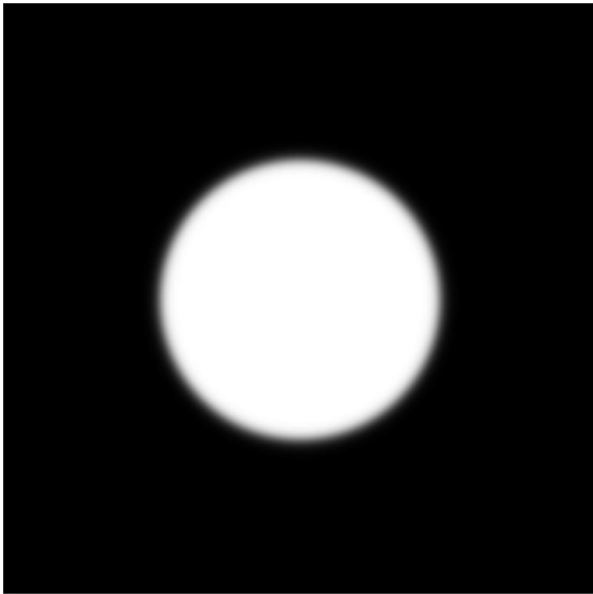


Fonte: Autoral

Analisando-se a Figura 10, é possível notar que os erros do tipo *ringing*, são imperceptíveis com o uso do filtro *butterworth*. Isso se dá pois, o filtro *butterworth* trabalha com transições suaves de frequência, porém, conforme a ordem do filtro aumenta, é possível notar o aparecimento de *ringing* na imagem.

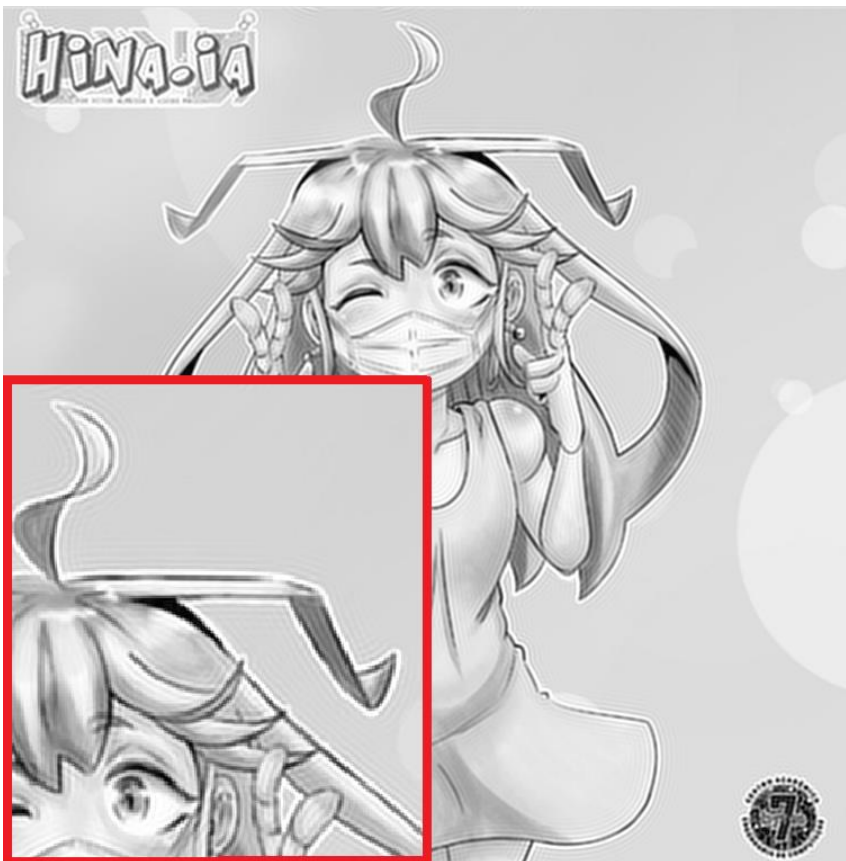
Como indicado em Gonsales,Wood(2010), os filtros *butterworth* passa-baixa, não apresentam valores negativos nem *ringing* para ordens n de até 2. Já, por outro lado, para ordens n altas como 20, o filtro *butterworth* passa a exibir características semelhante aos filtros circulares ideais. Para ilustrar essa ideia, a Figura 11 mostra a máscara de um BLPF com $n=20$ e a Figura 12 mostra a aplicação da máscara na imagem em questão.

Figura 11: Mascara do BLPF de $n=20$ e $D_o=180$



Fonte: Autoral

Figura 12: imagem filtrada com BLPF de $n=20$ e $D_o=180$



Fonte: Autoral

Observando a imagem da Figura 12, ao se aproximar é possível ver o surgimento de alguns ringings.

Código Questão 3:

```
%leitura da imagem
img1=imread('img1.jpg');
imshow(img1(:,:,:));
title('imagem original');
figure();

[x,y,z] = size(img1);

imgGray = rgb2gray(img1);

imshow(imgGray(:,:,:));
title('imagem em escala de cinza');
figure();

imgfft = fft2(imgGray);

imgfftShift = fftshift(imgfft);

imshow(log(abs(imgfftShift)),[]);
title('magnitudo da imagem');
figure();

%gerando mascara de filtro circular ideal passa baixa
Do=180; %diametro do circulo
n=2;    %ordem do filtro
masck = Butterworth_low_Pass_masck (img1,Do,n);

imshow(masck,[]);
title("mascara filtro butterworth passa-baixa ");
figure();

%realizando filtragem da imagem em magnitude de frequência
imgFiltred = imgfftShift .* masck;

imshow(log(abs(imgFiltred)),[]);
title('maginitude da imagem filtrada');
figure();

%voltando imagem para o domínio do tempo
imgFiltred_temp=uint8( abs ( ifft2(fftshift(imgFiltred) ) ));

imshow(imgFiltred_temp);
title('imagem filtrada BLPF com Do=' + string(Do)+' e n=' + string(n));
figure();

%gerando mascara de filtro butterworth
%essa função foi baseada nas equações expostas em
%(GONZALEZ, WOODS, 2010)pág 177
function masck = Butterworth_low_Pass_masck (img1,Do,n)

[x,y,~] = size(img1);

masck = zeros(x,y,'double');
% Do é o diâmentro do circulo, freq de corte
% Duv distancia entre o ponto e o centro
% para transições suaves masck(i,j) = (50% de 1) quando Duv=Do
for i=1: x
    for j=1:y
        Duv = (((i- x/2)^2 + (j - y/2)^2 )^(1/2));
        if( Duv == Do)
            masck(i,j)= 1*0.5;
        else
            masck(i,j)= 1/ ( 1+ ( Duv/Do)^(2*n) );
        end
    end
end
end
```

4. Carregue a imagem transforme em escala de cinza e execute a transformada, e em seguida crie uma máscara circular e aplique direto na magnitude da imagem escolhida para que seja executada a passa-alta ideal. Volte a imagem para o domínio do tempo e mostre e comente o resultado.

Resposta:

O filtro passa alta ideal (IHPF) é o oposto do filtro passa baixa ideal (ILPF), no sentido de que ele elimina todas as frequências dentro de um círculo de diâmetro D_0 , enquanto deixa passar, sem atenuação, todas as frequências fora do círculo. E, da mesma forma que o ILPF, o IHPF também apresenta fenômenos como efeito ringing. A Figura 13, mostra a função que define um filtro passa-alta ideal (GONZALEZ, WOODS, 2010).

Figura 13: Filtro passa-alta ideal (IHPF)

$$H(u,v) = \begin{cases} 0 & \text{se } D(u,v) \leq D_0 \\ 1 & \text{se } D(u,v) > D_0 \end{cases}$$

Fonte: (GONZALEZ, WOODS, 2010)

Onde:

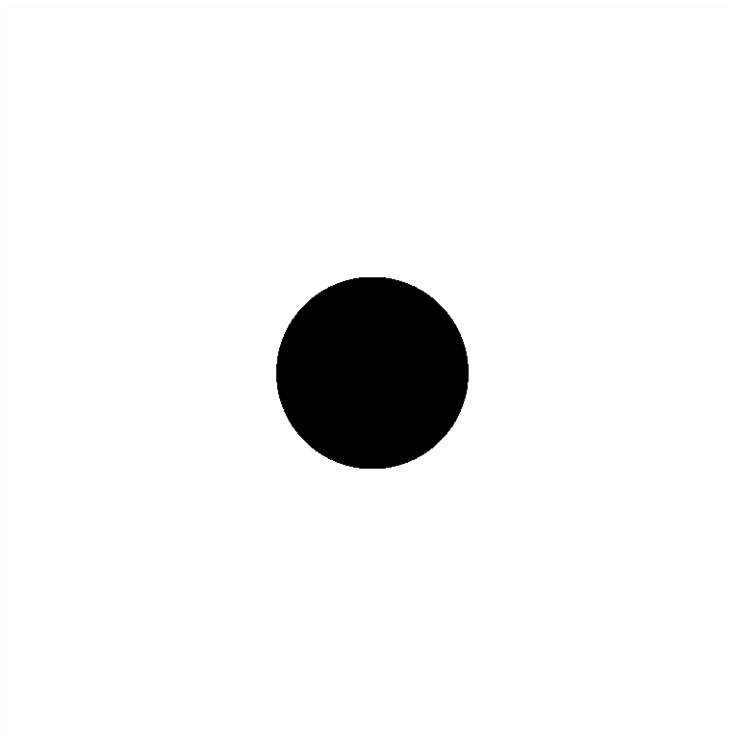
- D_0 : é a frequência de corte;
- $D(u,v)$ é a distância entre um ponto (u,v) e a origem da imagem, é dado por:

$$D(u,v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2} \quad .Eq1$$

- P e Q são as dimensões da máscara ($P \times Q$) (caso fosse uma operação de convolução estes tamanhos deveriam ser $P \geq 2M-1$ e $Q \geq 2N-1$).

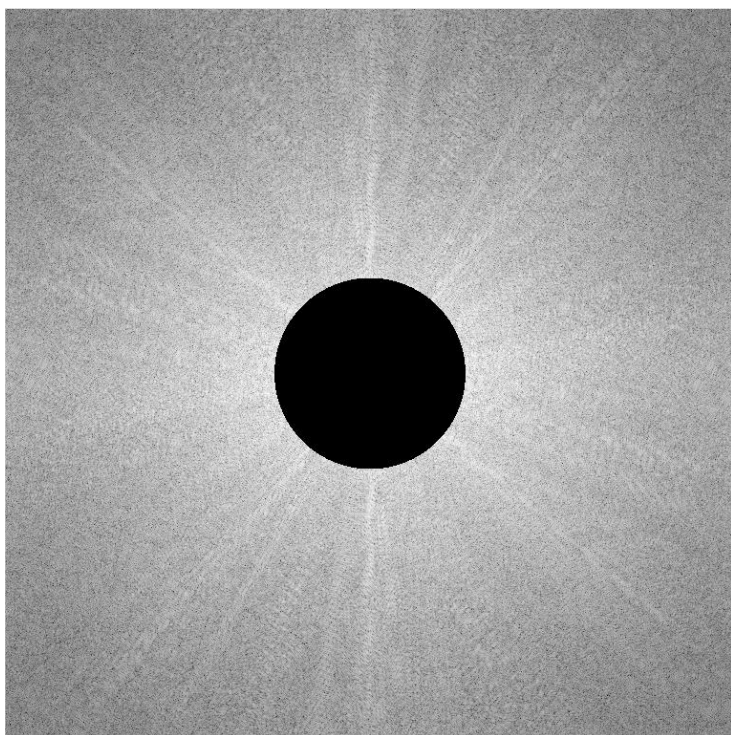
Deste modo, as Figuras 14, 15 e 16 mostram, para um $D_0=100$, a máscara do IHPF, a máscara aplicada direto na magnitude e a imagem filtrada com a máscara IHPF.

Figura 14: máscara do filtro passa alta ideal circular (IHPF) $D_0=100$



Fonte: Autoral

Figura 15: máscara IHPF aplicada na magnitude $D_0=100$



Fonte: Autoral

Figura 16: Imagem filtrada com IHPF de $D_0=100$



Fonte: Autoral

A partir da análise da Figura 16, é possível notar que, assim como mencionado anteriormente, a imagem apresentou ringing. Da mesma forma, é possível notar que o fundo da imagem ficou com valores 0, isso é devido a filtragem passa-alta ser análoga a diferenciação no domínio do espaço (GONZALEZ, WOODS, 2010).

Da mesma forma, também é possível notar que, o IHPF destacou com mais intensidade as bordas, esse destaque é reduzido conforme o tamanho de D_0 é aumentado, do mesmo jeito o ringing também é reduzido. Para tanto, a Figura 17, mostra a imagem filtra com um IHPF de $D_0=200$.

Figura 17: Imagem filtrada com IHPF de $D_0=200$



Fonte: Autoral

Comparando a Figura 16 com a Figura 17, é possível notar uma queda na quantidade de rínging, porém, também é possível notar que, a imagem ficou menos aparente. Isso se deve a quantidade de frequência cordada na Figura 17 com o $D_0=200$.

Código questão 4:

```
%leitura da imagem
img1=imread('img1.jpg');
imshow(img1(:,:,:));
title('imagem original');
figure();

[x,y,z] = size(img1);

imgGray = rgb2gray(img1);

imshow(imgGray(:,:,:));
title('imagem em escala de cinza');
figure();

imgfft = fft2(imgGray);

imgfftShift = fftshift(imgfft);

imshow(log(abs(imgfftShift)),[]);
title('magnitudo da imagem');
figure();

%gerando mascara de filtro circular ideal passa alta
Do=100; %diametro do circulo
masck = ideal_hight_Pass_Circular_masck(img1,Do);

imshow(masck(:,:,:).*255);
title("mascara filtro passa-alta ideal circular");
figure();

%realizando filtragem da imagem em magnitudo de frequência
imgFiltred = imgfftShift .* masck;

imshow(log(abs(imgFiltred)),[]);
title('maginitude da imagem filtrada');
figure();

%voltando imagem para o dominio do tempo
imgFiltred_temp=uint8( abs ( ifft2(fftshift(imgFiltred) ) ));
%imgFiltred_temp=uint8( ifft2(fftshift(imgFiltred) ));

imshow(imgFiltred_temp);
title('imagem filtrada ideal hight pass Do=' + string(Do));
figure();

imwrite(imgFiltred_temp,'imgQ5.jpg');

%gerando mascara de filtro circular ideal passa alta
%essa função foi baseada nas equações expostas em
%(GONZALEZ, WOODS, 2010)pág 184
function masck = ideal_hight_Pass_Circular_masck (img1,Do)

[x,y,z] = size(img1);

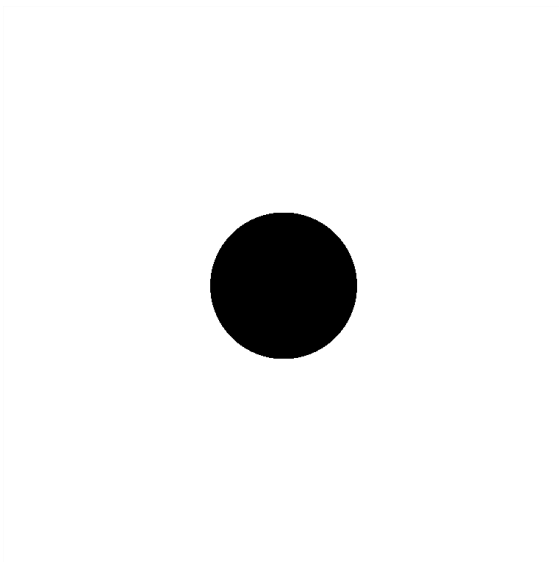
masck = zeros(x,y,'double');
% Do é o diâmentro do circulo, freq de corte
for i=1: x
    for j=1:y
        if( (( (i- x/2)^2 + (j - y/2)^2 )^(1/2)) >Do)
            masck(i,j)=1;
        end
    end
end
end
```

5. Utilize a imagem resultante na questão 4 e refaça-a questão com ela, explique se teve alguma diferença, e mostre a magnitude antes de aplicar a máscara de novo. Volte a imagem para o domínio do tempo e mostre o resultado.

Resposta:

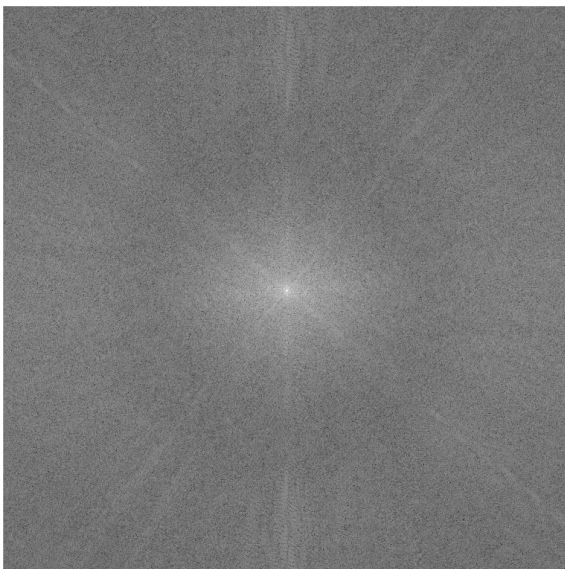
As Figura 18, 19, 20 e 21 mostram a máscara IHPF de $D_o=100$, a magnitude da imagem, a máscara aplicada na magnitude e a imagem resultando na questão 4 para um $D_o=100$ filtrada novamente com o mesmo $D_o=100$.

Figura 18: mascara IHPF $D_o=100$



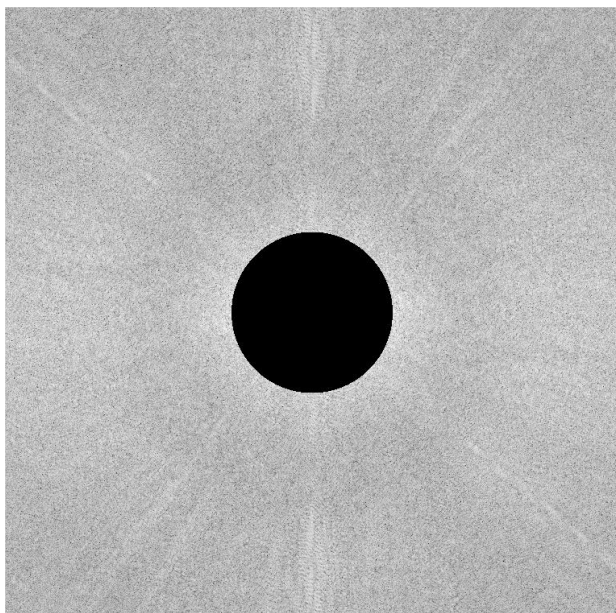
Fonte: Autoral

Figura 19: magnitude da imagem resultante em 4



Fonte: Autoral

Figura 20: mascara aplicada na magnitude



Fonte: Autoral

Figura 21: imagem resultante da questão 4 filtrada novamente para $D_0=100$



Fonte: Autoral

A partir da análise das Figuras 19, 20 e 21, é possível perceber os seguintes pontos:

- A magnitude da imagem na Figura 19 se comparada com a da Figura 4 e 15, é possível notar que a da Figura 19 é mais escura e mais atenuada, por conta de que, as frequências abaixo de $D_o=100$ na imagem resultante da questão 4, já estavam cortadas;
- A aplicação da máscara direto na magnitude vista na Figura 20, deu uma aparência mais clara, isso pode se dever a nova filtragem que cortou mais uma vez magnitudes de intensidades que estão fora de $D_o=100$;
- A imagem final, mostrada na Figura 21, ficou com os traços menos aparentes do que a da Figura 16, isso se deve a questão da imagem já ter vindo com valores de frequência cortados em $D_o \leq 100$, deste modo, o próximo cálculo do filtro para $D_o=100$, provocou um visual menos aparente dos traços da imagem, sendo que o fundo, continuou preto.

Código questão 5:

```
%leitura da imagem
img1=imread('imgQ5.jpg');
imshow(img1(:,:,:));
title('imagem original');
figure();

[x,y,z] = size(img1);

% não se usa rgb2gray(img1) porque a imagem já esta em escala de cinza
imgGray = img1;

imshow(imgGray(:,:,:));
title('imagem em escala de cinza');
figure();

imgfft = fft2(imgGray);

imgfftShift = fftshift(imgfft);

imshow(log(abs(imgfftShift)),[]);
title('magnitudo da imagem');
figure();

%gerando mascara de filtro circular ideal passa alta
Do=100; %diametro do circulo
masck = ideal_hight_Pass_Circular_masck(img1,Do);

imshow(masck(:,:,:).*255);
title("mascara filtro passa-alta ideal circular");
figure();

%realizando filtragem da imagem em magnitude de frequência
imgFiltred = imgfftShift .* masck;

imshow(log(abs(imgFiltred)),[]);
title('maginitude da imagem filtrada');
figure();

%voltando imagem para o domínio do tempo
imgFiltred_temp=uint8( abs ( ifft2(fftshift(imgFiltred) ) ));
%imgFiltred_temp=uint8( ifft2(fftshift(imgFiltred) ));

imshow(imgFiltred_temp);
title('imagem filtrada ideal hight pass Do=' + string(Do));
figure();

%gerando mascara de filtro circular ideal passa alta
%essa função foi baseada nas equações expostas em
%(GONZALEZ, WOODS, 2010)pág 184
function masck = ideal_hight_Pass_Circular_masck (img1,Do)

[x,y,~] = size(img1);

masck = zeros(x,y,'double');
% Do é o diâmentro do circulo, freq de corte
for i=1: x
    for j=1:y
        if( (( (i- x/2)^2 + (j - y/2)^2 )^(1/2)) >Do)
            masck(i,j)=1;
        end
    end
end
end
```


6. Refaça questão 2 com uma máscara retangular, dimensões a sua escolha. Explique a diferença visual com a imagem resultante da questão 2.

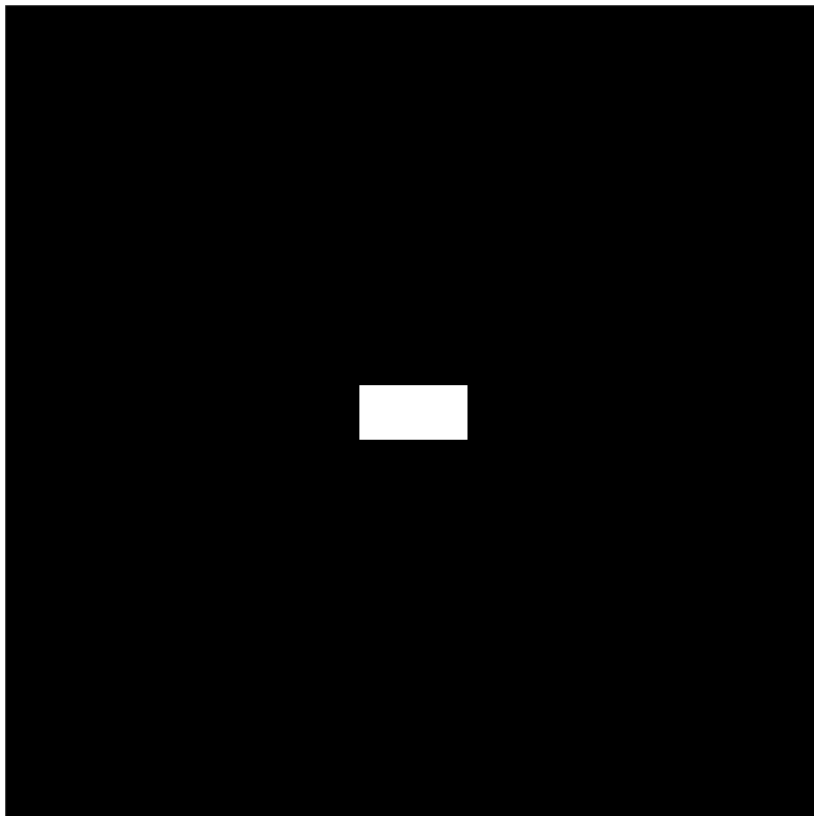
Resposta:

Foi decidido por se produzir um filtro retangular sem rotação, isto é, sua base está sobre o eixo horizontal. Para tanto, também foi criado um filtro, o qual, o ponto de centralização escolhido foi a origem (centro da imagem). As dimensões da mascara são definidas por $DI=50$ (numero de linhas) e $Dc=100$ (número de colunas).

Escolhido o tipo de máscara, foi notado que, na imagem de exemplo que consta na lista disponibilizada, foi utilizado um filtro passa alta retangular, ao invés de um filtro passa baixa retangular, como enunciado na questão 2. Optei por permanecer com o filtro passa baixa, só que dessa vez utilizando mascara retangular, como dito no enunciado da questão 6.

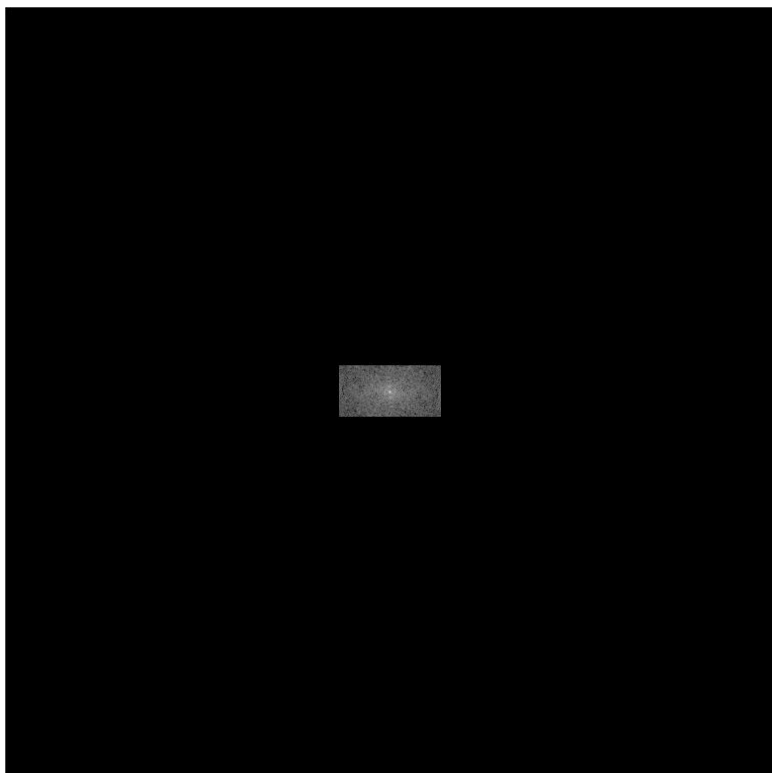
Deste modo, as Figuras 22, 23 e 24 mostram a mascara definida, a mascara aplicada na magnitude e a imagem final filtrada utilizando a máscara.

Figura 22: Mascara filtro passa-baixa retangular $DI=50$, $Dc=100$



Fonte: Autoral

Figura 23: Mascara aplicada na magnitude



Fonte: Autoral

Figura 23: imagem filtrada com filtro passa-baixa retangular $Dl=50$, $Dc=100$



Fonte: Autoral

Analisando as Figuras anteriores, é possível notar que, o resultado final foi uma imagem bem mais desfocada e cheia de ringing que as obtidas na questão 2. Isso se dá por conta de que, o filtro retangular, ao contrário da circular, deixa de pegar frequência de forma uniforme em alguns ângulos intermediários, então seus cortes são mais abruptos e menos precisos.

Por conta deste comportamento, é preciso se criar uma janela maior para o filtro. Para visualizar melhor esta afirmação, a Figura 24 mostra a imagem filtrada com uma máscara de janela de dimensões $DI=200$ e $Dc=400$.

Figura 24: imagem filtrada com filtro passa baixa retangular $DI=200$ e $Dc=400$



Fonte: Autoral

A visualização da imagem melhorou bastante, se assemelhando a da questão 2, isso porque, com dimensões 200×400 a máscara é capaz de coletar mais frequência em mais ângulos diferentes. Também se nota, ao se mudar as dimensões da máscara somente em um sentido, a presença de uma distorção ortogonal ao da escala aplicada. As Figuras 25 e 26 mostram este efeito para duas máscaras de dimensões diferentes.

Figura 25: Filtro passa- baixa retangular $DI=50$, $Dc=400$



Fonte: Autoral

Figura 26: Filtro passa- baixa retangular $DI=400$, $Dc=50$



Fonte: Autoral

Nas Figuras 25 e 26, percebe-se uma distorção mais acentuada do tipo *ringing* em 2 posições diferentes, vertical e horizontal. Esse efeito, é devido a baixa coleta de informações de frequência nessas regiões por conta das dimensões da máscara, o que resulta em regiões menos definidas que outras.

Código questão 6:

```
%leitura da imagem
img1=imread('img1.jpg');
imshow(img1(:,:));
title('imagem original');
figure();

[x,y,z] = size(img1);

imgGray = rgb2gray(img1);

imshow(imgGray(:,:));
title('imagem em escala de cinza');
figure();

imgfft = fft2(imgGray);

imgfftShift = fftshift(imgfft);

imshow(log(abs(imgfftShift)),[]);
title('magnitudo da imagem');
figure();

%gerando mascara de filtro circular ideal passa alta
Dl=200; %Altura do quadrado em linhas
Dc=400; %Largura do retangulo em colunas
masck = ideal_low_Pass_rect_masck(img1,Dl,Dc);

imshow(masck(:,:).*255);
title("mascara filtro passa-baixa ideal retangular Dl=" + string(Dl)+ " x Dc=" +string(Dc));
figure();

%realizando filtragem da imagem em magnitudo de frequência
imgFiltred = imgfftShift .* masck;

imshow(log(abs(imgFiltred)),[]);
title('maginitude da imagem filtrada');
figure();

%voltando imagem para o dominio do tempo
imgFiltred_temp=uint8( abs ( ifft2(fftshift(imgFiltred) ) ));
imgFiltred_temp=uint8( ifft2(fftshift(imgFiltred) ) );

imshow(imgFiltred_temp);
title('imagem filtrada ideal low pass retagular Dl=' + string(Dl)+ ' x Dc=' +string(Dc));
figure();

%gerando mascara de filtro retangular ideal passa baixa
%essa função foi baseada nas equações expostas em
%(GONZALEZ, WOODS, 2010)pág 184
%Dl e Dc são as dimensões do retangulo em linhas por colunas respectivamente
function masck = ideal_low_Pass_rect_masck(img1,Dl,Dc)

[x,y,~] = size(img1);

masck = zeros(x,y,'double');
% Dx1 e Dx2 são o calculo da altura do retangulo em relação a origem
% Dy1 e Dy2 são o calculo da largura do retangulo em relação a origem
Dx1= round( (x/2)-(Dl/2) );
Dx2= round( (x/2)+(Dl/2) );
Dy1= round( (y/2)-(Dc/2) );
Dy2= round( (y/2)+(Dc/2) );

%construindo a mascara retangular
for i= Dx1:Dx2
    for j= Dy1:Dy2
        masck(i,j)=1;
    end
end
end
```

7. Execute a questão 2 mas a imagem resultante deve ser colorida.

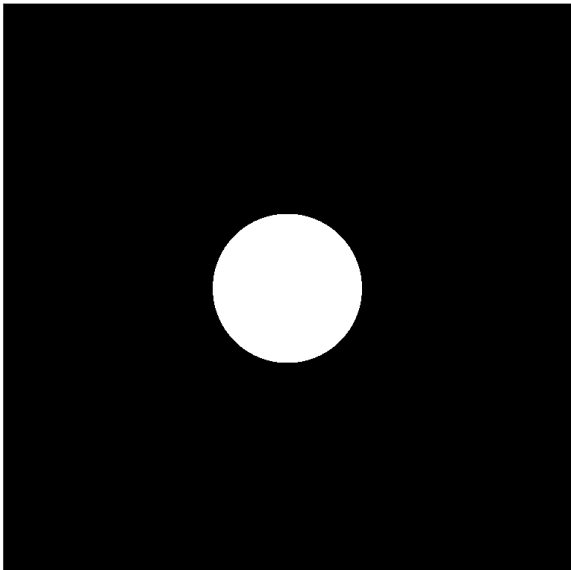
Resposta:

Para a execução da filtragem utilizando o filtro passa-baixa ideal circular, na imagem colorida, foi necessário aplicar as operações em cada componente de cor R, G e B separadamente. Deste modo, foram realizados os seguintes procedimentos em cada um dos canais de cores:

- Transformada rápida de Fourier 2D – **fft2()**;
- Passagem das magnitudes da extremidade para o centro – **fftshift()**;
- Criação da máscara *ideal low-pass*;
- Produto da máscara com cada componente R, G e B separadamente;
- Transformada inversa de Fourier para voltar ao domínio do tempo — **ifft2()**;
- Mostrar imagem filtrada na tela — **imshow()**;

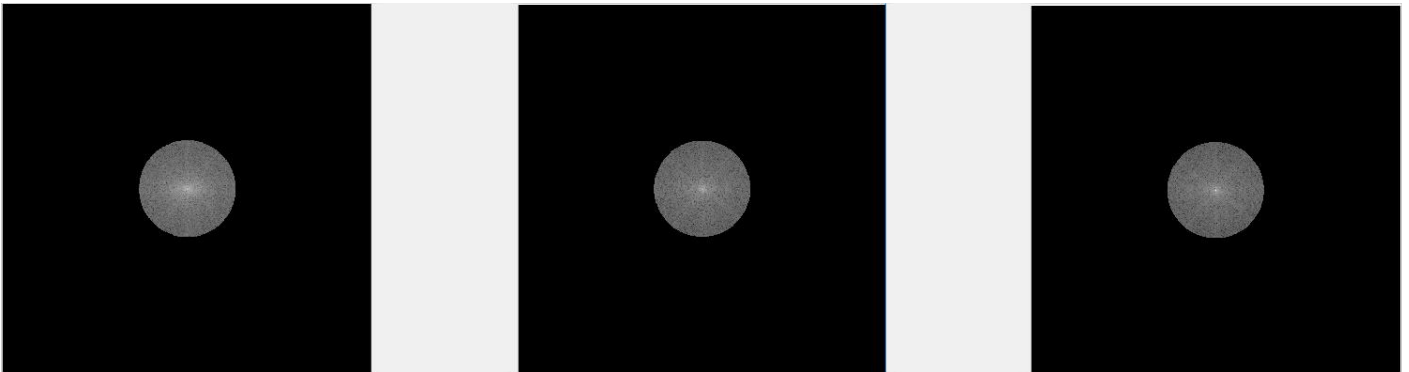
Dado os pontos indicados acima, as Figuras 26, 27 e 28 mostram a máscara, as operações de filtragem na magnitude em R, G e B e a imagem colorida filtrada, destacando que o Do continuou como 100.

Figura 26: mascara filtro ideal passa-baixa circular, Do=100



Fonte: Autoral

Figura 27: mascara direta na magnitude em R, G e B $D_0=100$



Fonte: Autoral

Figura 28: imagem colorida filtrada com *ideal low-pass filter*, $D_0=100$



Fonte: Autoral

Comparando-se a Figura 6 com a Figura 28, é possível notar que, os efeitos de *ringing* também apareceram na imagem colorida, assim como, também houve um borramento. De modo geral, o efeito do filtro na imagem colorida foi o equivalente ao efeito do mesmo na imagem em tons de cinza, inclusive, as cores foram mantidas fieis a figura original.

Código questão 7:

```
%leitura da imagem
img1=imread('img1.jpg');
imshow(img1(:,:,));
title('imagem original');
figure();

[x,y,z] = size(img1);
% rgb2gray(img1) não é necessário converter a imagem para cinza
imgGray = img1(:,:,);

imgfft=zeros(x,y,z,'double');
imgfft(:,:,1) = fft2(imgGray(:,:,1));
imgfft(:,:,2) = fft2(imgGray(:,:,2));
imgfft(:,:,3) = fft2(imgGray(:,:,3));

imgfftShift=zeros(x,y,z,'double');
imgfftShift(:,:,1) = fftshift(imgfft(:,:,1));
imgfftShift(:,:,2) = fftshift(imgfft(:,:,2));
imgfftShift(:,:,3) = fftshift(imgfft(:,:,3));

%gerando mascara de filtro circular ideal passa baixa
Do=100; %diametro do circulo
masck = ideal_low_Pass_Circular_masck(img1,Do);
imshow(masck(:,:,).*255);
title("mascara filtro passa-baixa ideal circular");
figure();

%realizando filtragem da imagem em magnitude de frequência em R, G e B
imgFiltred=zeros(x,y,z,'double');
imgFiltred(:,:,1) = imgfftShift(:,:,1) .* masck;
imgFiltred(:,:,2) = imgfftShift(:,:,2) .* masck;
imgFiltred(:,:,3) = imgfftShift(:,:,3) .* masck;

imshow(log(abs(imgFiltred(:,:,1))),[]);
title('maginitude da imagem filtrada R');
figure();

imshow(log(abs(imgFiltred(:,:,2))),[]);
title('maginitude da imagem filtrada G');
figure();

imshow(log(abs(imgFiltred(:,:,3))),[]);
title('maginitude da imagem filtrada B');
figure();

%voltando imagem para o dominio do tempo
imgFiltred_temp=zeros(x,y,z,'uint8');
imgFiltred_temp(:,:,1)=uint8( abs ( ifft2(fftshift(imgFiltred (:,:,1)) ) ));
imgFiltred_temp(:,:,2)=uint8( abs ( ifft2(fftshift(imgFiltred (:,:,2)) ) ));
imgFiltred_temp(:,:,3)=uint8( abs ( ifft2(fftshift(imgFiltred (:,:,3)) ) ));

imshow(imgFiltred_temp);
title('imagem filtrada ideal low pass Do=' + string(Do));
figure();

%gerando mascara de filtro circular ideal passa baixa
%essa função foi baseada nas equações expostas em
%(GONZALEZ, WOODS, 2010)pág 176
function masck = ideal_low_Pass_Circular_masck (img1,Do)

[x,y,~] = size(img1);

masck = zeros(x,y,'double');
% Do é o diâmentro do circulo, freq de corte
for i=1: x
    for j=1:y
        if( (( (i- x/2)^2 + (j - y/2)^2 )^(1/2)) <=Do)
            masck(i,j)=1;
        end
    end
end
end
```

REFERÊNCIAS

GONZALEZ, Rafael C., WOODS, Richard E. **Processamento Digital de Imagem**. São Paulo: Sv, 2010.