

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO LUIZ GRAVE GROSS
180171

Relatório – Laboratório 3

Trabalho da Disciplina de Fundamentos de
Processamento de Imagens

Prof. Manuel Menezes de Oliveira Neto

Porto Alegre, 26 de outubro de 2011.

Questão 1. Download as imagens “high_pass.bmp” e “low_pass.bmp” dos seguintes endereços:

- http://www.inf.ufrgs.br/~oliveira/high_pass.bmp
- http://www.inf.ufrgs.br/~oliveira/low_pass.bmp
- <http://www.inf.ufrgs.br/~oliveira/zebra.jpg>
- http://www.inf.ufrgs.br/~oliveira/Periodic_noise_Clown.tif

Salve-as no diretório “work” do MATLAB.

O diretório de trabalho foi outro. O comando a seguir especifica o caminho de trabalho e já garante o acesso ao diretório em questão.

```
>> cd C:\Users\João\ Gross\Dropbox\UFRGS\2011-02\INF01046' - Fundamentos de  
Processamento de Imagens\Laboratorio3\
```

Questão 2. Escreva um procedimento para:

(a) Ler e exibir a imagem low_pass.bmp utilizando os comando imread e imshow, vistos anteriormente.

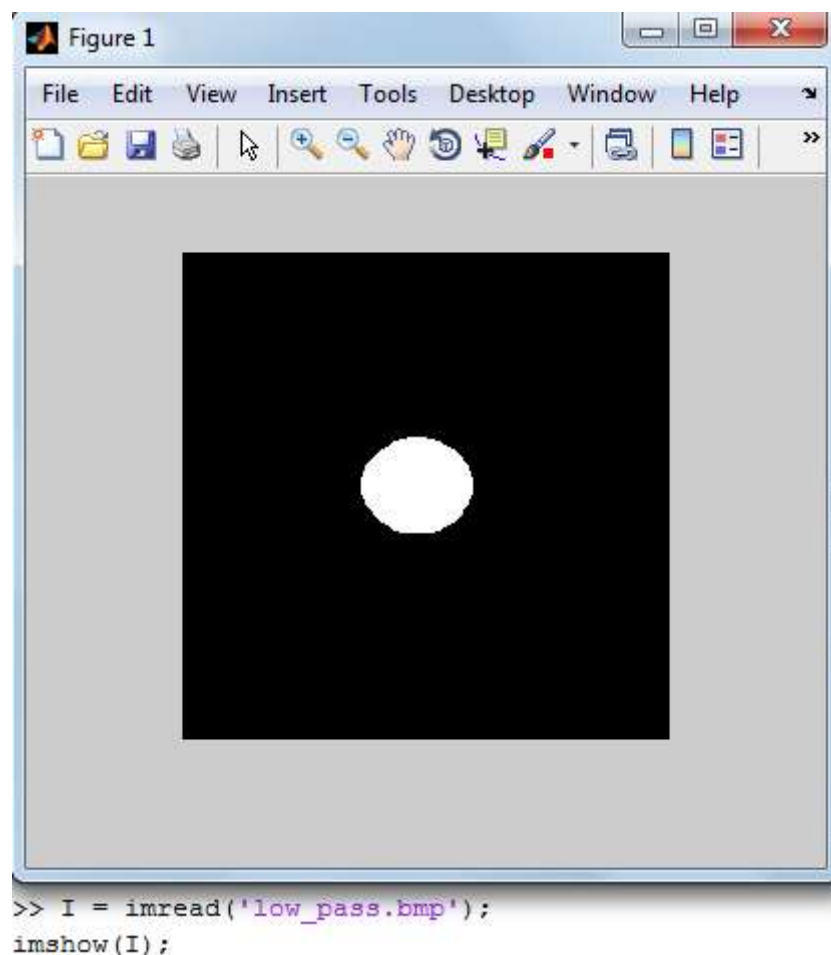


Figura 1: imagem low_pass.bmp

(b) Os pixels desta imagem contém intensidades com valores 0 (preto) e 255 (branco). Substituir os valores “255” por “1” (e.g., divida os valores da máscara por 255, $m = m / 255$;). Ao realizar esta conversão, você terá preparado uma máscara para um filtro (passa baixas) a ser implementado no domínio frequência.

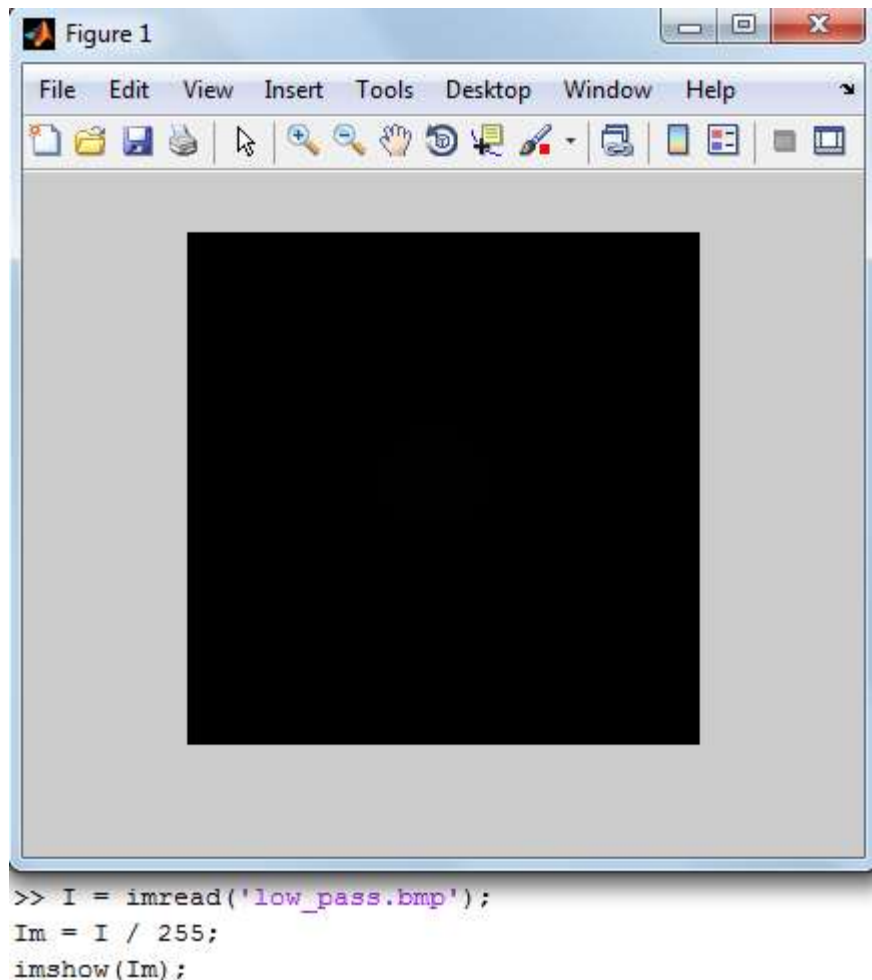


Figura 2: Imagem com tons setados para 0 e 1

Na Figura 2 não conseguimos ver a diferença entre os pixels com valor 0 e 1, porém comprovei que os comandos tiveram resultado, pois executando `imshow(Im)`, sem o ponto e vírgula, a matriz correspondente da imagem só possuía 0's e 1's.

(c) Ler a imagem do cameraman e calcular sua transformada de Fourier (comando `fft2`);

Trivial, como já havia sido feito no laboratório 2.

```
>> C = imread('cameraman.tif');
>> Cf = fft2(C);
```

(d) Visualize o resultado utilizando `imshow(log(abs()), [3, 10])`. A seguir, aplique um “shift” (comando `fftshift`) à imagem transformada e visualize o resultado.

```
>> C = imread('cameraman.tif');
>> Cf = fft2(C);
>> subplot(1,2,1); imshow(log(abs(Cf)), [3, 10]);
>> Cfs = fftshift(Cf);
>> subplot(1,2,2); imshow(log(abs(Cfs)), [3, 10]);
```

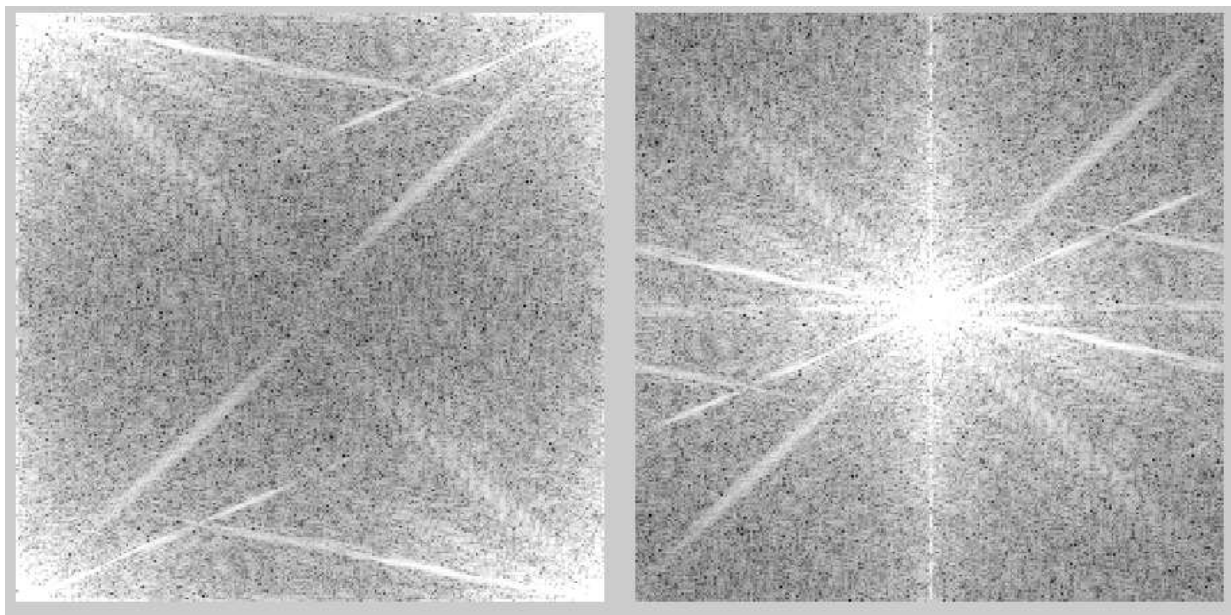


Figura 3: Espectros de amplitude. À esquerda o espectro original e à direita o espectro com shift

(e) Multiplique a transformada (após o shift) pela mascara (após a substituição de 255s por 1s) utilizando o operador “.” (ponto asterisco – multiplicação elemento a elemento das duas matrizes) e exiba o resultado (não esqueça de usar o log...).

Observação:

(1) Algumas versões do Matlab não permitem o uso do operador “.” entre números complexos (resultado da transformada) e números inteiros (máscara). Neste caso, você precisará realizar um “cast” da mascara para double: Ex.: $M = \text{img transf shift} .* \text{double}(\text{img mascara})$;

```
>> I = imread('low_pass.bmp');
>> Im = I / 255;
>> C = imread('cameraman.tif');
>> Cfs = fftshift(fft2(C));
>> M = Cfs .* double(Im);
>> imshow(log(abs(M)), [3, 10]);
```

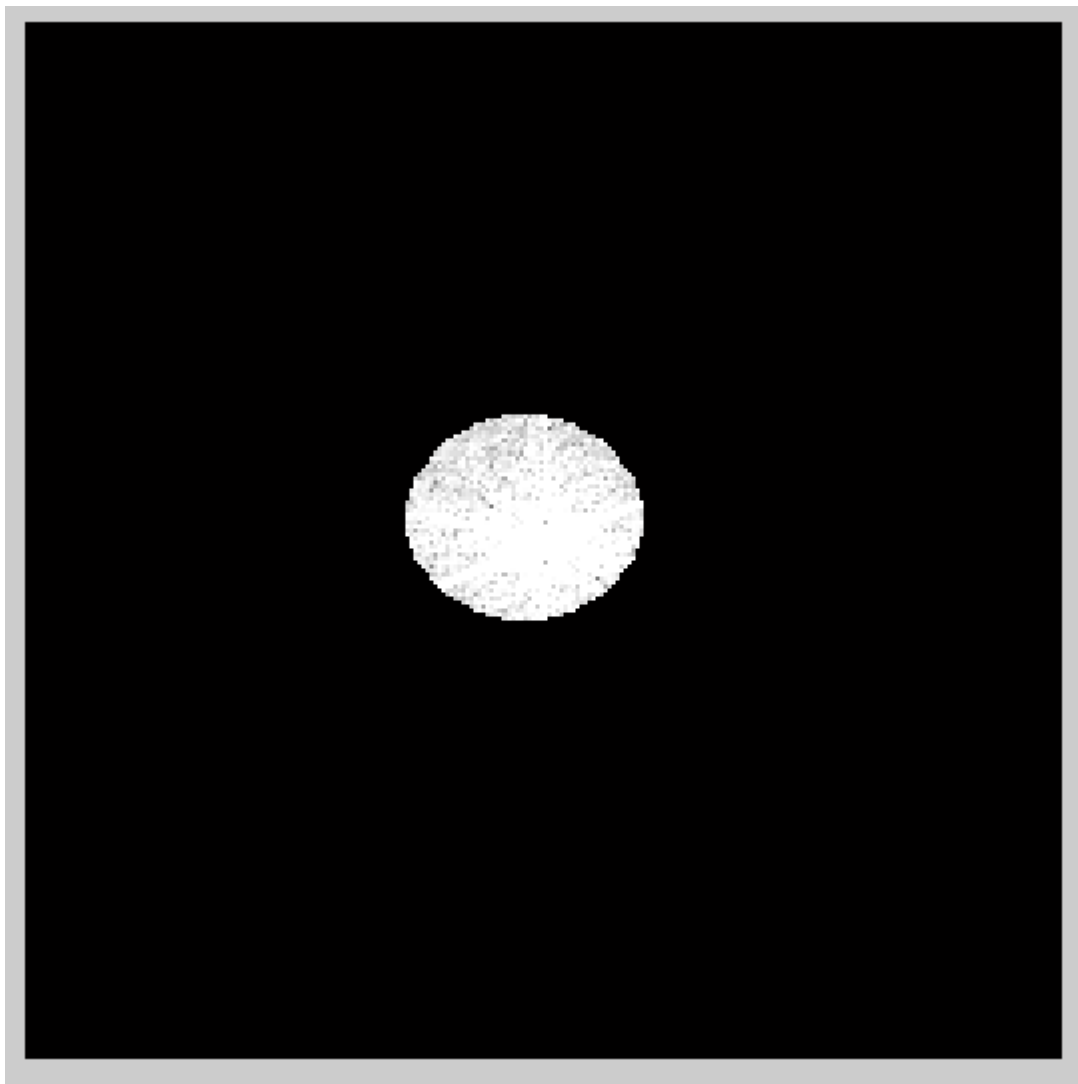


Figura 4: Aplicação da máscara (filtro passa baixa) no resultado da transformada de Fourier com shift

(f) Applique o operador shift ao resultado obtido no passo (e) e exiba do resultado.

```
>> Ms = fftshift(M);  
>> imshow(log(abs(Ms)), [3, 10]);
```

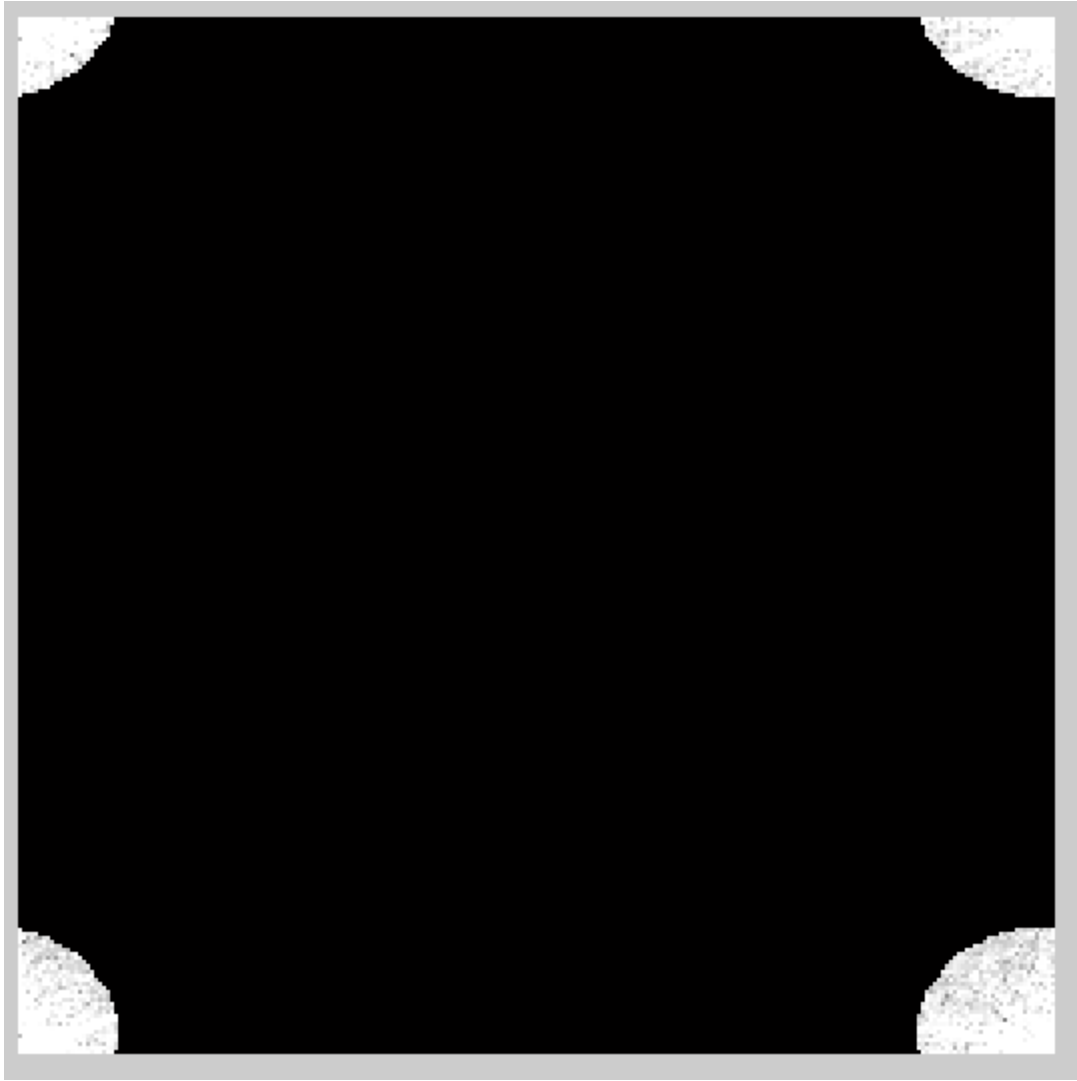


Figura 5: Figura 4 com a aplicação de shift

(g) Aplique a transformada inversa (`ifft2`) à imagem obtida no passo (f) e exiba o resultado (não esqueça de converter o resultado para `uint8` antes de exibir).

```
>> Msi = ifft2(Ms);  
>> imshow(uint8(Msi));
```



Figura 6: cameraman.tif com a aplicação do filtro passa baixa

(h) Explique o que aconteceu em cada um dos passos anteriores.

Acompanhe a seguir a descrição das etapas realizadas na questão 2:

(a): filtro passa baixa é carregado em uma matriz;

(b): valores 255 do filtro passa baixa são passados para 1. Isso fará sentido na etapa (e);

(c): calculada a transformada de Fourier da imagem cameraman.tif;

(d): shift da transformada de Fourier obtida em (c);

(e): multiplicação entre a máscara e a transformada de Fourier obtida em (d). Aqui é que ocorre a aplicação do filtro passa baixa. Agora a operação realizada em (b) aqui faz sentido, pois estando só a região central do filtro (máscara) em 1 e a transformada de Fourier centralizada com as menores frequências, apenas essas frequências mais baixas do centro do espectro de amplitude se mantém na transformada, as demais, por estarem sendo multiplicadas por 0, são excluídas da transformada;

(f): realizado um novo shift sobre a matriz obtida em (e). Isso é necessário para que possamos reconstruir a imagem cameraman.tif, visto que precisamos anular o efeito do shift realizado em (d);

(g): reconstrução da imagem cameraman.tif. Percebemos que a imagem está borrada, pois as frequências mais altas (transições bruscas entre tons) foram eliminadas da imagem pela aplicação do filtro passa baixa na transformada. Devemos observar que este mesmo efeito pode ser obtido pela operação de convolução no domínio espacial, enquanto no domínio frequência, basta a

multiplicação ponto a ponto do filtro pelo espectro de amplitudes e depois reconstruir a imagem no domínio espacial.

Questão 3. Repita os passos descritos no item 2 para a imagem `high_pass.bmp`.

(a) Ler e exibir a imagem `low_pass.bmp` utilizando os comando `imread` e `imshow`, vistos anteriormente.

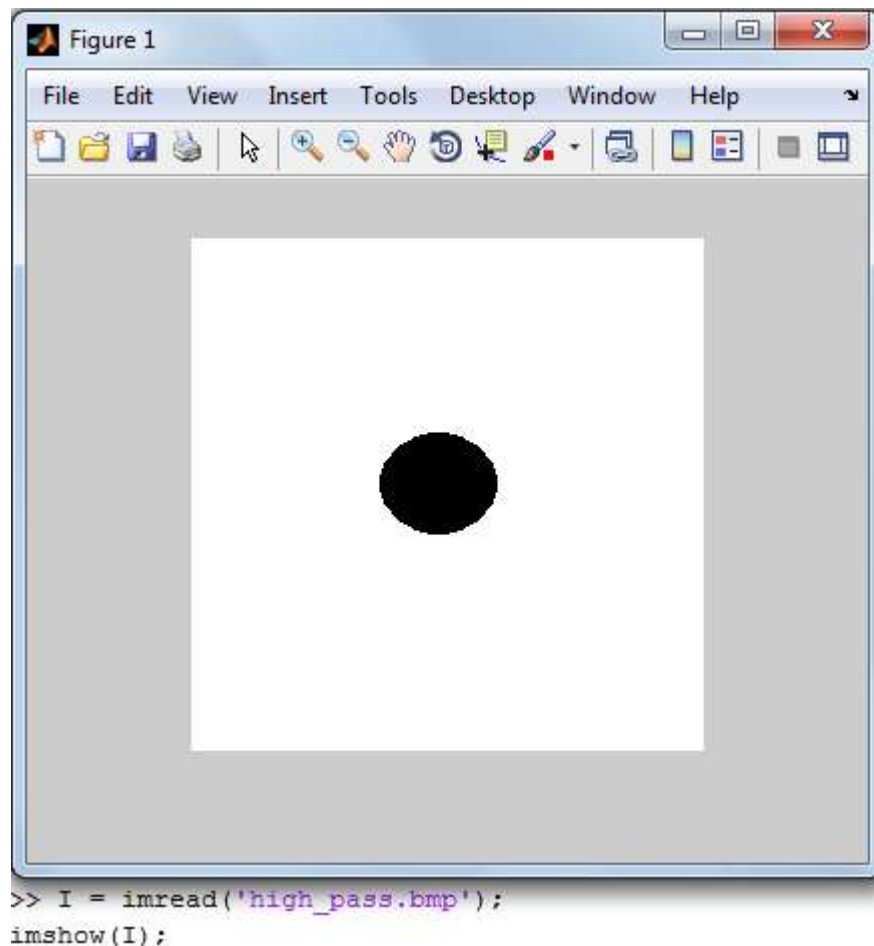


Figura 7: Filtro passa alta

(b) Os pixels desta imagem contém intensidades com valores 0 (preto) e 255 (branco). Substituir os valores “255” por “1” (e.g., divida os valores da máscara por 255, $m = m / 255$;). Ao realizar esta conversão, você terá preparado uma máscara para um filtro (passa baixas) a ser implementado no domínio frequência.

Ídem ao item (b) da questão 2.

(c) Ler a imagem do cameraman e calcular sua transformada de Fourier (comando `fft2`);

Ídem ao item (c) da questão 2.

(d) Visualize o resultado utilizando `imshow(log(abs()), [3, 10])`. A seguir, aplique um “shift” (comando `fftshift`) à imagem transformada e visualize o resultado.

Ídem ao item (d) da questão 2.

(e) Multiplique a transformada (após o shift) pela máscara (após a substituição de 255s por 1s) utilizando o operador “.” (ponto asterisco – multiplicação elemento a elemento das duas matrizes) e exiba o resultado (não esqueça de usar o log...).

Observação:

(1) Algumas versões do Matlab não permitem o uso do operador “.” entre números complexos (resultado da transformada) e números inteiros (máscara). Neste caso, você precisará realizar um “cast” da máscara para double: Ex.: $M = \langle \text{img transf shift} \rangle .* \text{double}(\langle \text{img mascara} \rangle);$

```
>> I = imread('high_pass.bmp');  
>> Im = I / 255;  
>> C = imread('cameraman.tif');  
>> Cfs = fftshift(fft2(C));  
>> M = Cfs .* double(Im);  
>> imshow(log(abs(M)), [3, 10]);
```

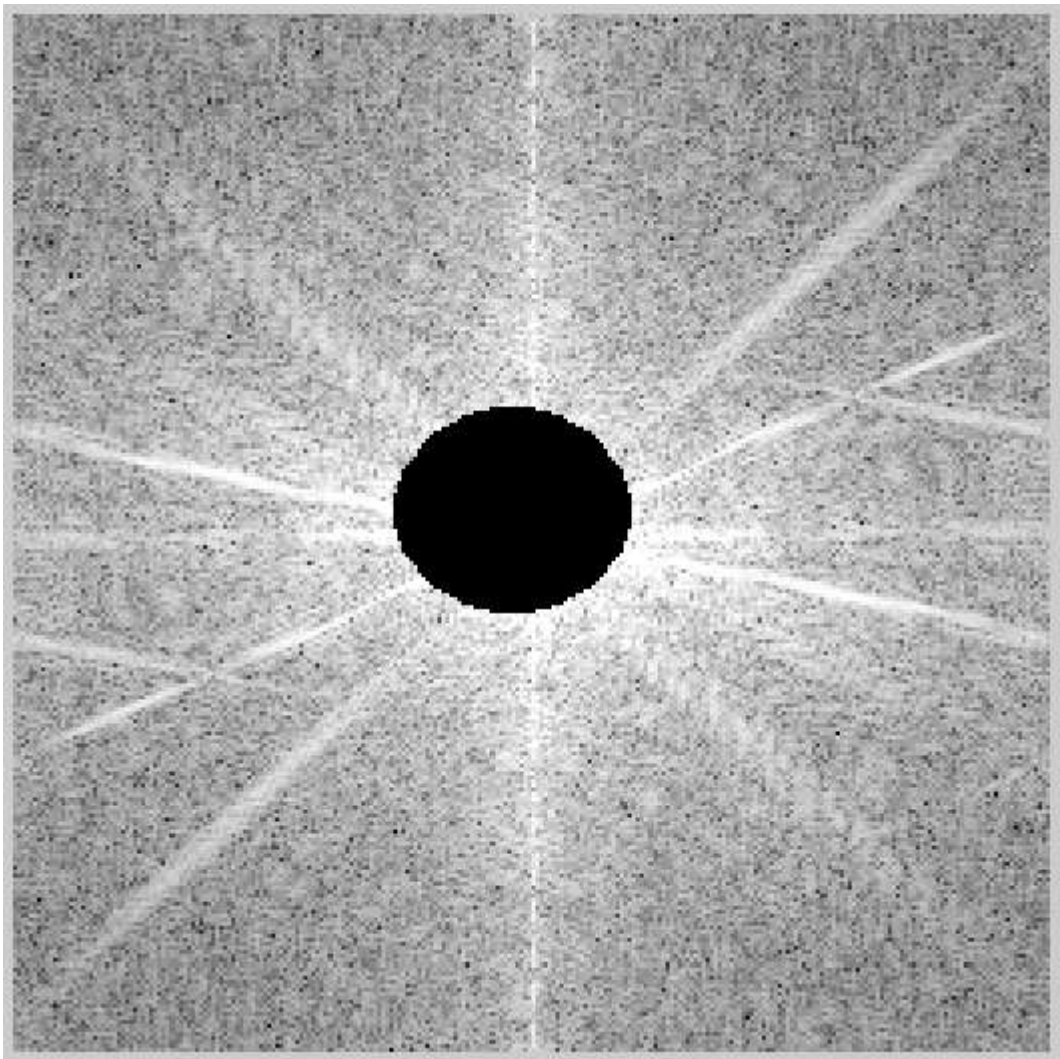


Figura 8: Aplicação da máscara (filtro passa alta) no resultado da transformada de Fourier com shift

(f) Aplique o operador shift ao resultado obtido no passo (e) e exiba do resultado.

```
>> Ms = fftshift(M);  
>> imshow(log(abs(Ms)), [3, 10]);
```

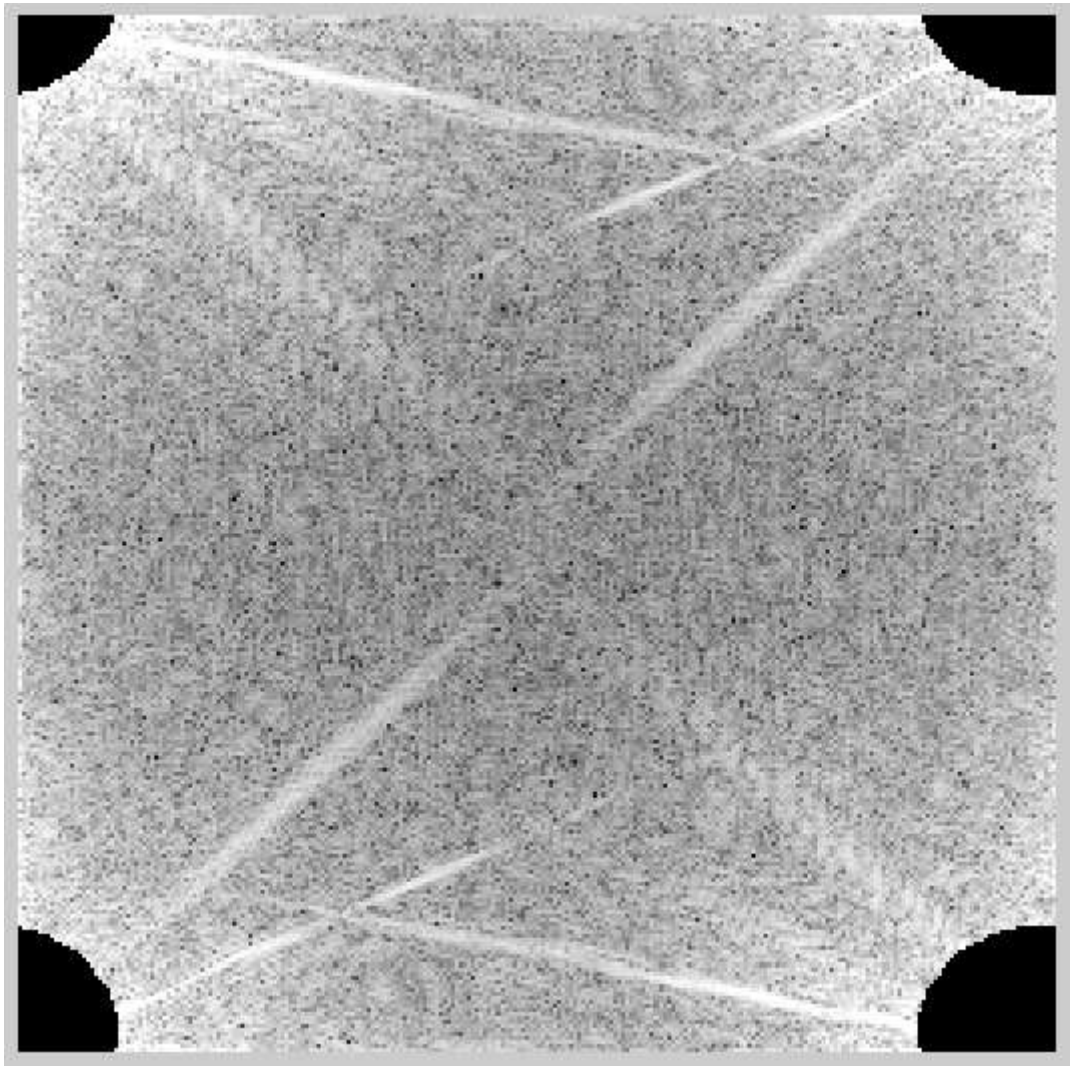


Figura 9: Figura 8 com a aplicação de shift

(g) Aplique a transformada inversa (ifft2) à imagem obtida no passo (f) e exiba o resultado (não esqueça de converter o resultado para uint8 antes de exibir).

```
>> Msi = ifft2(Ms);  
>> imshow(uint8(Msi));
```



Figura 10: cameraman.tif com a aplicação do filtro passa alta

(h) Explique o que aconteceu em cada um dos passos anteriores.

Muito semelhante à descrição de passos da questão 2; acompanhe a descrição dos passos da questão 3:

- (a): filtro passa alta é carregado em uma matriz;
- (b): valores 255 do filtro passa alta são passados para 1. Isso fará sentido na etapa (e);
- (c): calculada a transformada de Fourier da imagem cameraman.tif;
- (d): shift da transformada de Fourier obtida em (c);
- (e): multiplicação entre a máscara e a transformada de Fourier obtida em (d). Aqui é que ocorre a aplicação do filtro passa alta. Agora a operação realizada em (b) aqui faz sentido, pois estando só a região central do filtro (máscara) em 0 e a transformada de Fourier centralizada com as menores frequências, apenas essas frequências mais baixas do centro do espectro de amplitude serão eliminadas na transformada, as demais, por estarem sendo multiplicadas por 1, são se mantem;
- (f): realizado um novo shift sobre a matriz obtida em (e). Isso é necessário para que possamos reconstruir a imagem cameraman.tif, visto que precisamos anular o efeito do shift realizado em (d);
- (g): reconstrução da imagem cameraman.tif. Percebemos que a imagem está com os contornos ressaltados, pois as frequências mais baixas (tons de preenchimento dos objetos) foram eliminadas da imagem pela aplicação do filtro passa alta na transformada, ficando apenas os tons de transições bruscas, ou seja, as bordas dos objetos. Devemos observar que este mesmo efeito pode

ser obtido pela operação de convolução no domínio espacial, enquanto no domínio frequência, basta a multiplicação ponto a ponto do filtro pelo espectro de amplitudes e depois reconstruir a imagem no domínio espacial.

Questão 4. Observando o conteúdo da imagem zebra.jpg, faça um esboço de como você imagina ser o seu espectro de amplitude desta imagem, justificando o seu raciocínio. Somente depois de realizar o esboço e a justificativa, calcule e exiba tal espectro. Compare-o com o seu esboço tente explicar eventuais diferenças entre o previsto e o obtido.

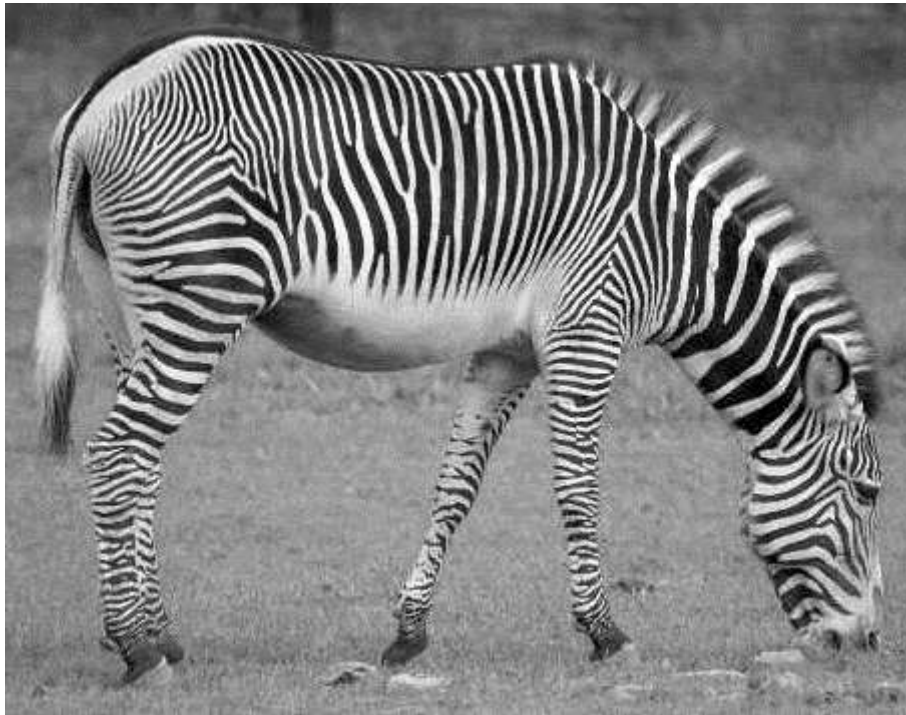


Figura 11: Imagem zebra.jpg

Como a imagem zebra.jpg possui muitas transições bruscas de cores, imaginei em um espectro de amplitude com bastantes frequências altas, ou seja, muitos pontos brancos.

```
>> I = imread('zebra.jpg');  
>> Fs = fftshift(fft2(I));  
>> imshow(log(abs(Fs)), [3, 10]);
```

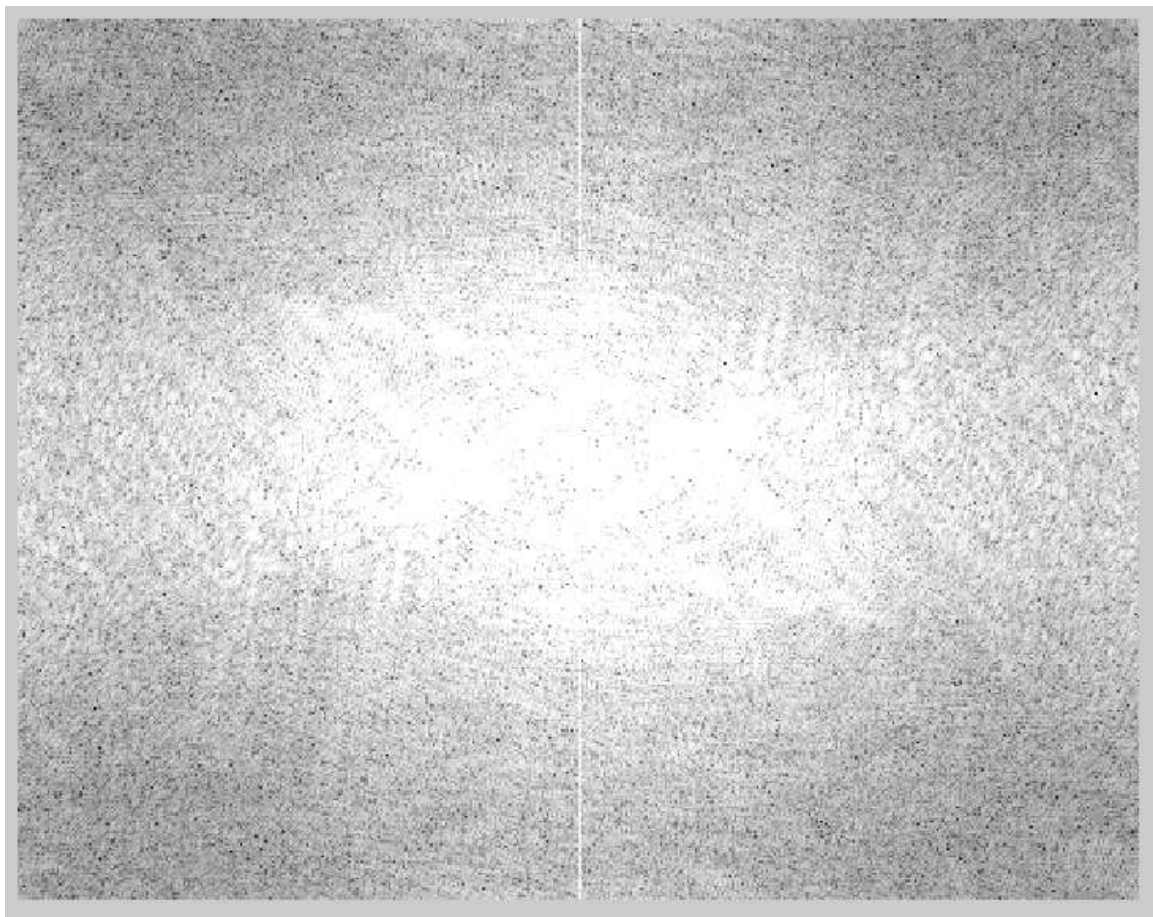


Figura 12: Espectro de amplitude da imagem zebra.jpg

Como podemos observar o centro do espectro, onde estão as menores frequências, está cercado de grandes grupos de altas frequências, devido à quantidade excessiva de transições bruscas de tons na imagem zebra.jpg.

Questão 5. Construa uma máscara para remoção do ruído periódico presente na imagem Periodic_noise_Clown.tif e realize a filtragem da imagem.

Observações:

- (1) A máscara deve conter as mesmas dimensões que a imagem que você quer filtrar;
- (2) Uma imagem da máscara apesar de binária (preto e branco) pode conter três canais (R,G, B). Você pode verificar o número de canais da imagem utilizando o comando `size(<imagem>)`;
- (3) Você pode gerar uma imagem binária com um único canal utilizando o programa “Paint” do Windows e salvando a imagem no formato .bmp com a opção “Monochrome Bitmap”.

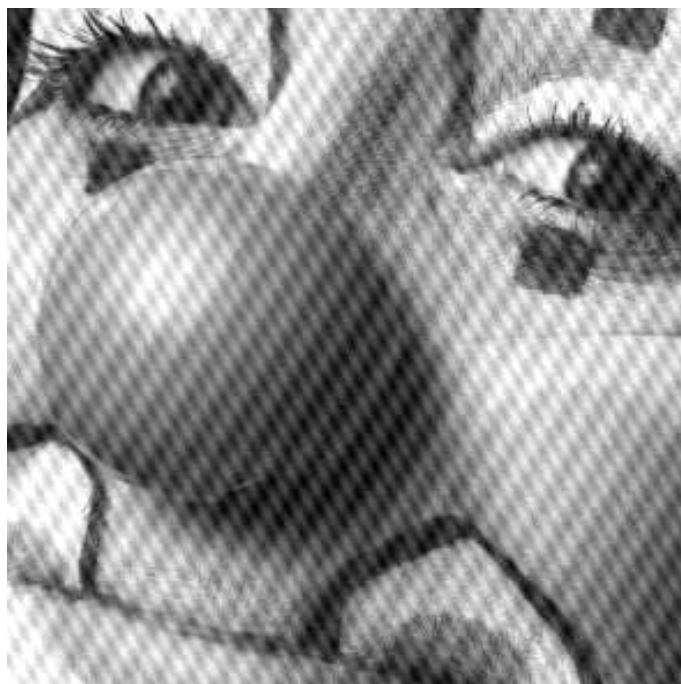


Figura 13: Figura Periodic_noise_Clown.tif

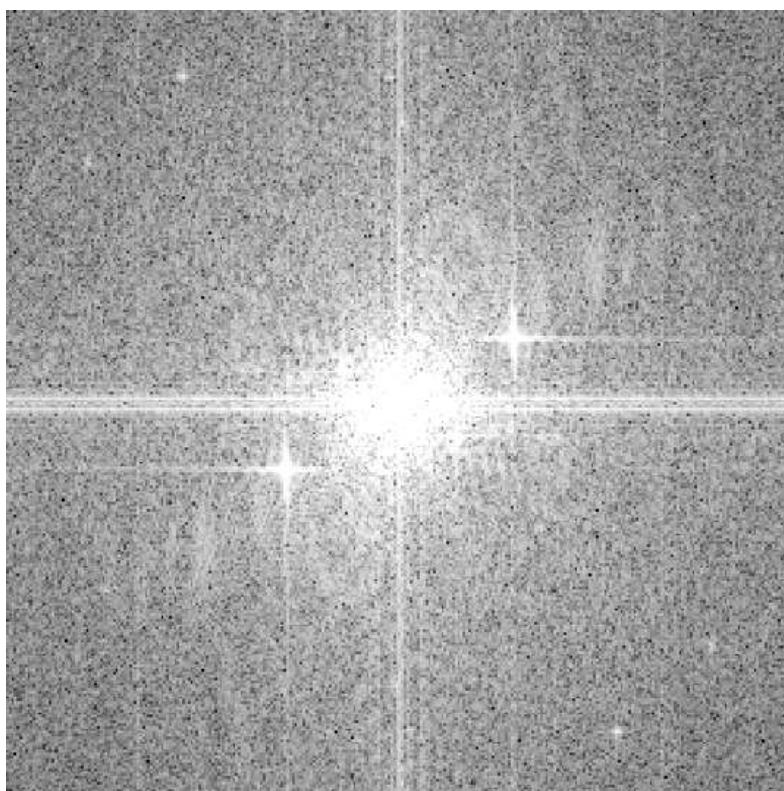


Figura 14: Espectro de amplitude da imagem
Periodic_noise_Clown.tif com shift

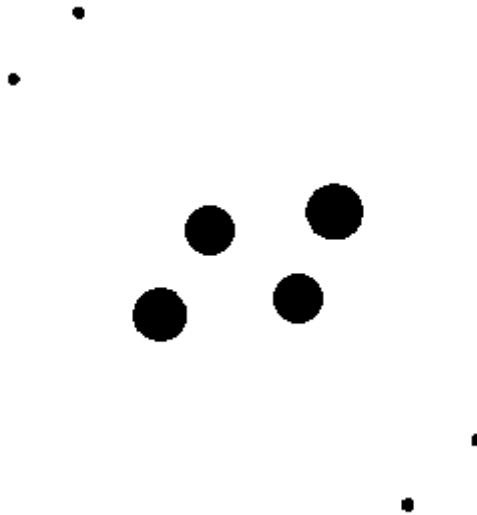


Figura 15: Máscara

```
>> clown = imread('Periodic_noise_Clown.tif');
>> clown_fourier_shift = fftshift(fft2(clown));
>> mask = imread('5_mascara.bmp');
>> clown_fourier_shift_mask = clown_fourier_shift .* double(mask);
>> imshow(log(abs(clown_fourier_shift_mask)), [3, 10]);
```

O resultado do trecho de código acima pode ser visualizado na Figura 16.

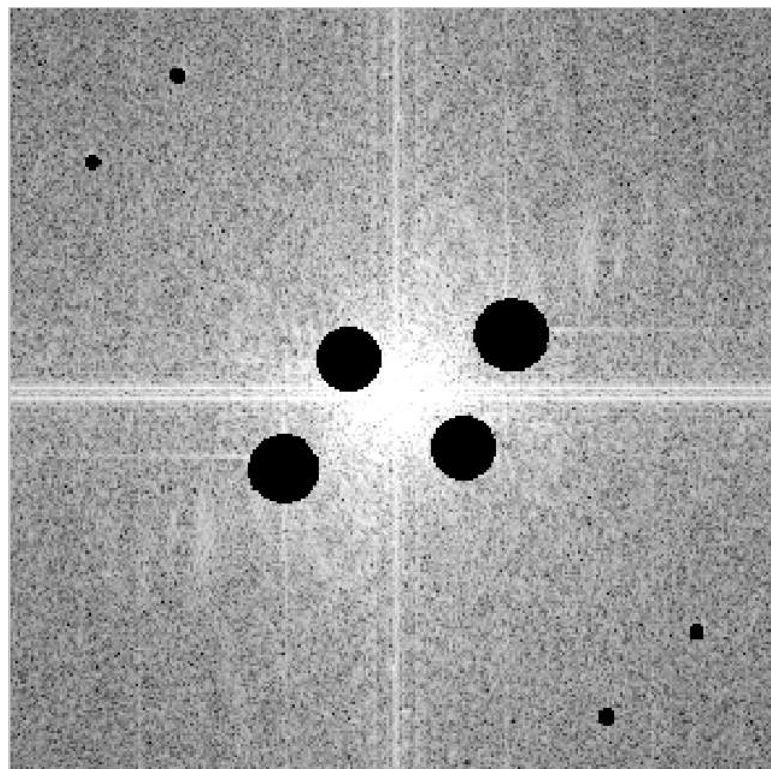


Figura 16: Espectro com aplicação da máscara


```
>> clown_fourier_mask = fftshift(clown_fourier_shift_mask);  
>> clown_mask = ifft2(clown_fourier_mask);  
>> imshow(uint8(clown_mask));
```



Figura 17: Figura Periodic_noise_Clown.tif após a aplicação da máscara na transformada de Fourier

Na Figura 17 fica evidente o efeito do filtro, com a remoção do ruído periódico da Figura 13.

Questão 6. Altere a(s) máscara(s) que você gerou para o Exercício 5 de modo a remover, separadamente, apenas os coeficientes associados às frequências $F(u,v)$, e depois apenas os coeficientes associados às frequências $F(-u,-v)$ do ruído periódico (veja exemplo de espectro filtrado para o caso $F(u,v)$ na imagem abaixo). O que acontece neste caso? Compare o seu resultado com a imagem original (ainda contendo o ruído periódico) e com a imagem filtrada obtida no Exercício 5.

$F(u,v)$:



Figura 18: Máscara para $F(u,v)$

```
>> clown = imread('Periodic_noise_Clown.tif');
>> clown_fourier_shift = fftshift(fft2(clown));
>> mask = imread('6_mascara_f(u,v).bmp');
>> clown_fourier_shift_mask = clown_fourier_shift .* double(mask);
>> imshow(log(abs(clown_fourier_shift_mask)), [3, 10]);
```

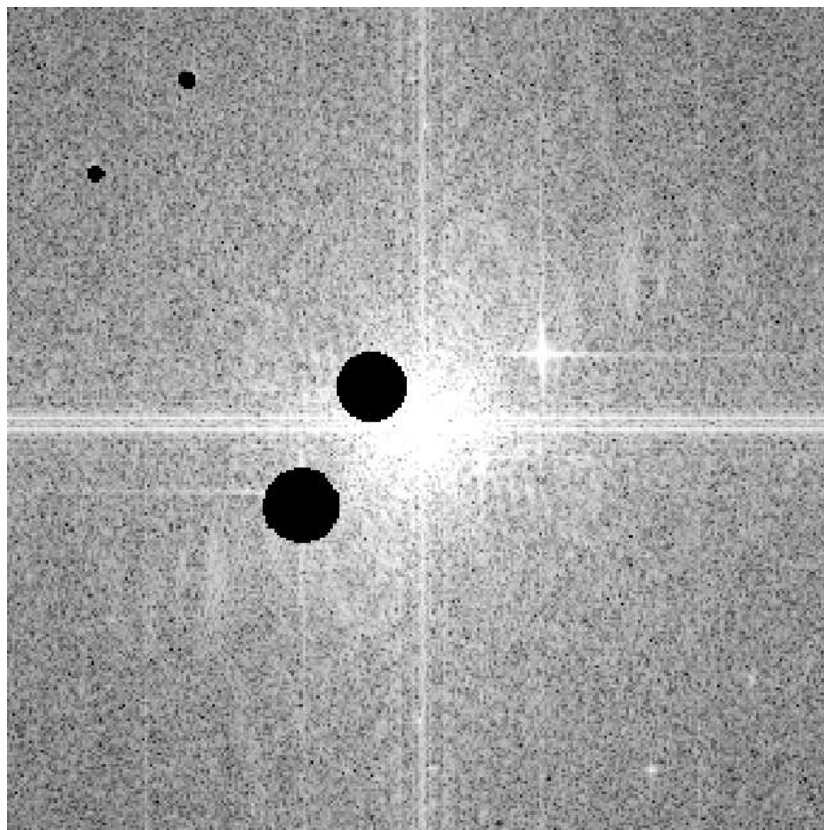


Figura 19: Espectro de amplitude com a máscara da Figura 18

```
>> clown_fourier_mask = fftshift(clown_fourier_shift_mask);  
>> clown_mask = ifft2(clown_fourier_mask);  
>> imshow(uint8(clown_mask));
```



Figura 20: Periodic_noise_Clown.tif com máscara $F(u,v)$

$F(-u,-v)$:



Figura 21: Máscara para $F(-u,-v)$

```
>> clown = imread('Periodic_noise_Clown.tif');  
>> clown_fourier_shift = fftshift(fft2(clown));  
>> mask = imread('6_mascara_f(-u,-v).bmp');  
>> clown_fourier_shift_mask = clown_fourier_shift .* double(mask);  
>> imshow(log(abs(clown_fourier_shift_mask)), [3, 10]);
```

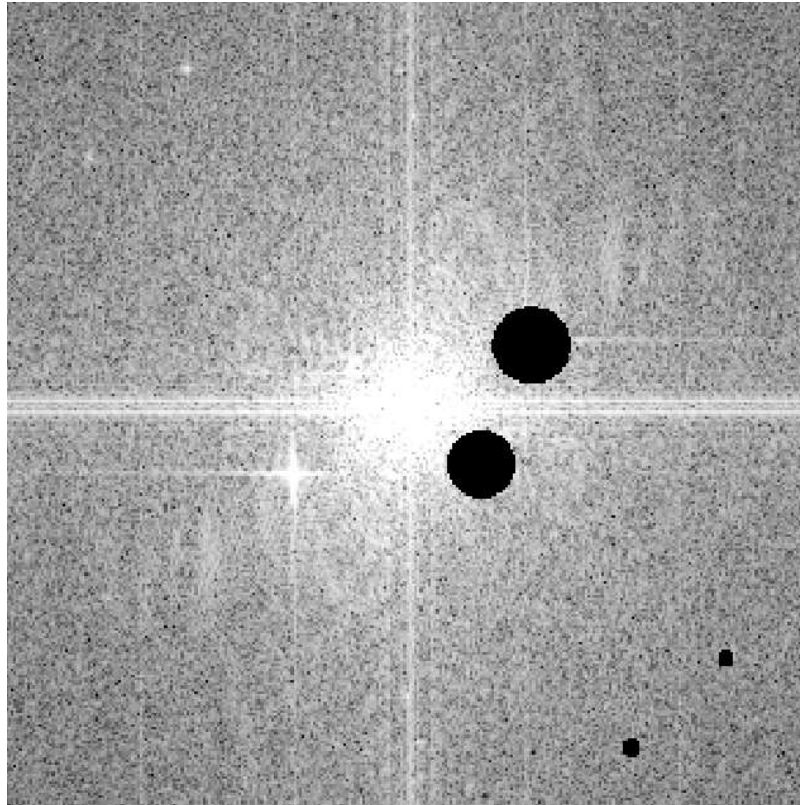


Figura 22: Espectro de amplitude com a máscara da Figura 21

```
>> clown_fourier_mask = fftshift(clown_fourier_shift_mask);  
>> clown_mask = ifft2(clown_fourier_mask);  
>> imshow(uint8(clown_mask));
```

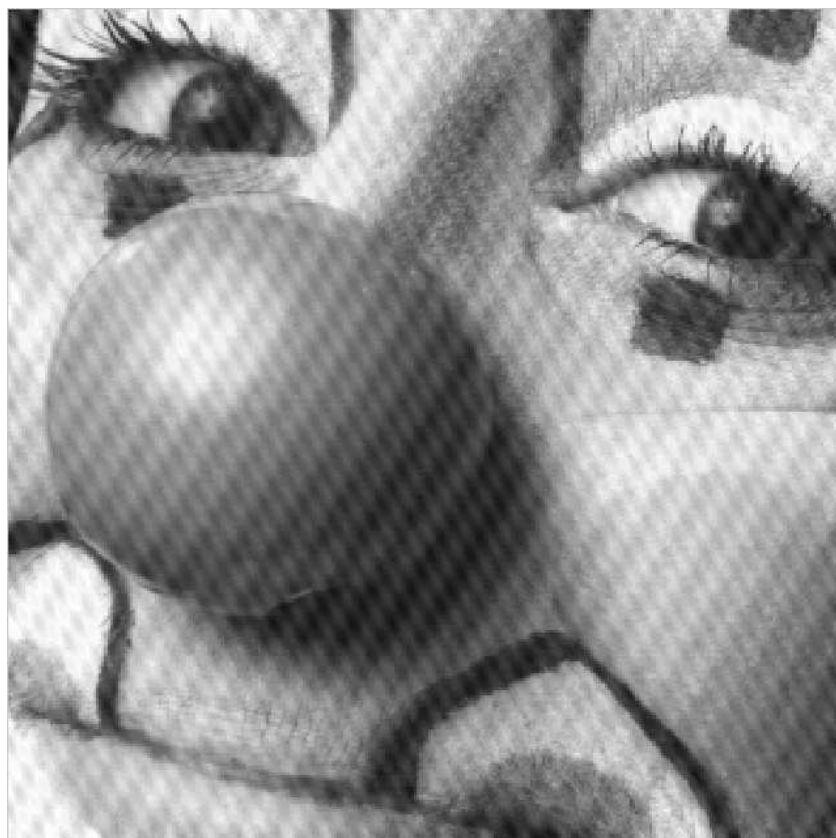


Figura 23: Periodic_noise_Clown.tif com máscara $F(-u,-v)$

Tanto a Figura 20, quanto a Figura 23, ou seja, a imagem Periodic_noise_Clown.tif com a máscara $F(u,v)$ e $F(-u,-v)$ respectivamente, possuem um ruído menor do que a Periodic_noise_Clown.tif original, contudo ainda não são tão 'limpas', quanto a imagem do resultado da questão 5, pois ainda restam regiões de ambos espectros de amplitude com frequências altas não filtradas.