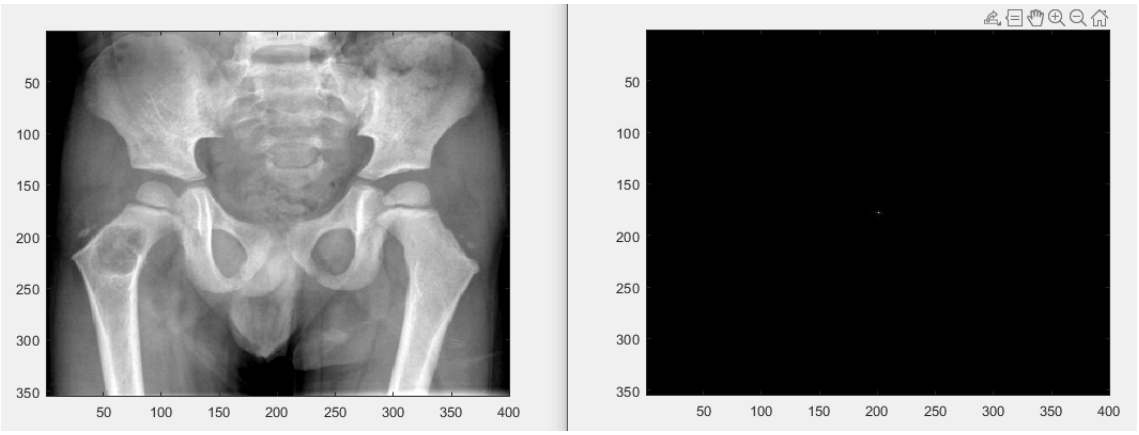
5910173 – Princípios de Imagens Médicas

Aula prática 7

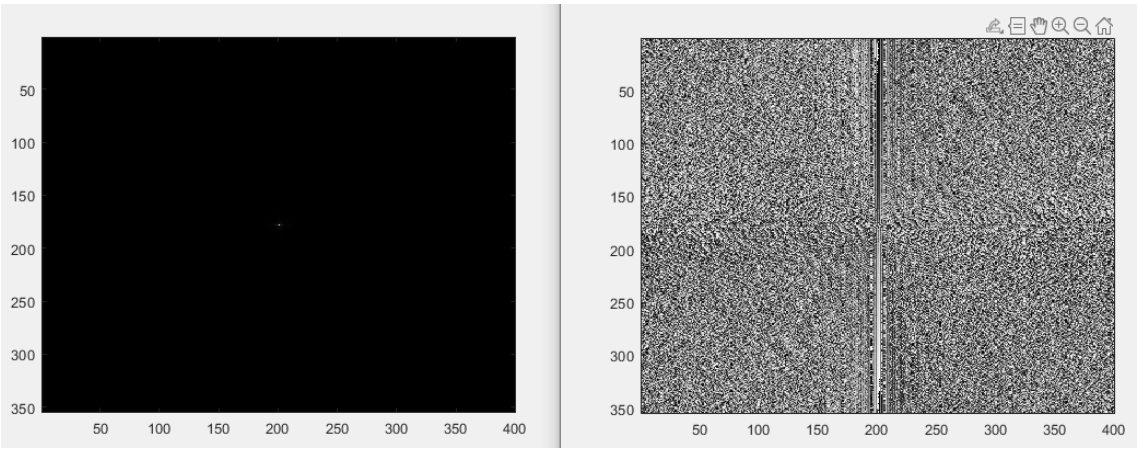
Imagens no espaço de Fourier - Filtros em frequência

Este roteiro será usado para as aulas: Transformada de Fourier e Filtros no domínio da Frequência. Salve as respostas dos exercícios em um único arquivo com o nome: **NOME_p7.doc**.

1. Escreva uma rotina para gerar imagens no espaço-k a partir da transformada de Fourier 2D. Dica: use as funções “fft2” e “fftshift”.



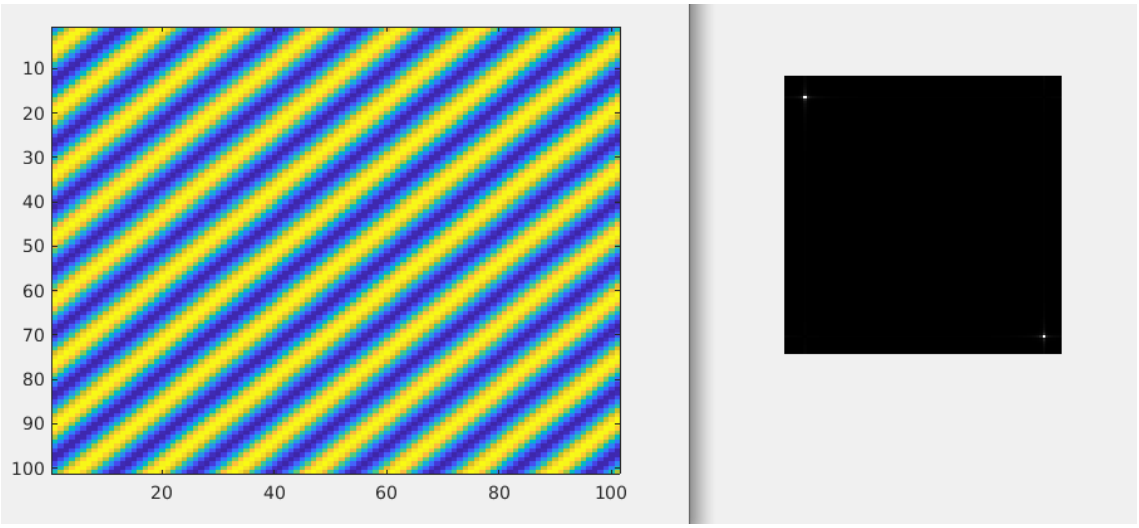
2. Mostre as imagens de magnitude e fase no espaço-k de PelvisRadiography.jpg.



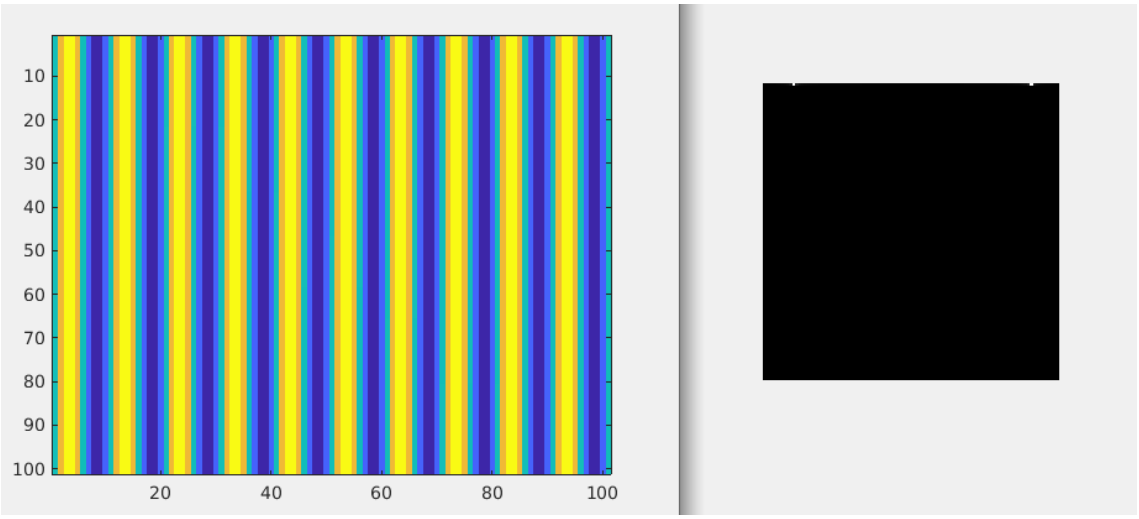
3. O código 1 ao final do roteiro cria uma grade senoidal. Com esse código é possível variar a frequência e o ângulo dessa grade. Teste o código no Matlab/Octave e tente entender sua lógica. A partir desse código, mostre e analise as imagens de magnitude no espaço de Fourier para cinco grades senoidais diferentes. Use pelo menos três ângulos e duas frequências.

```
%  Ângulo de rotação
sinephas = pi/4;
```

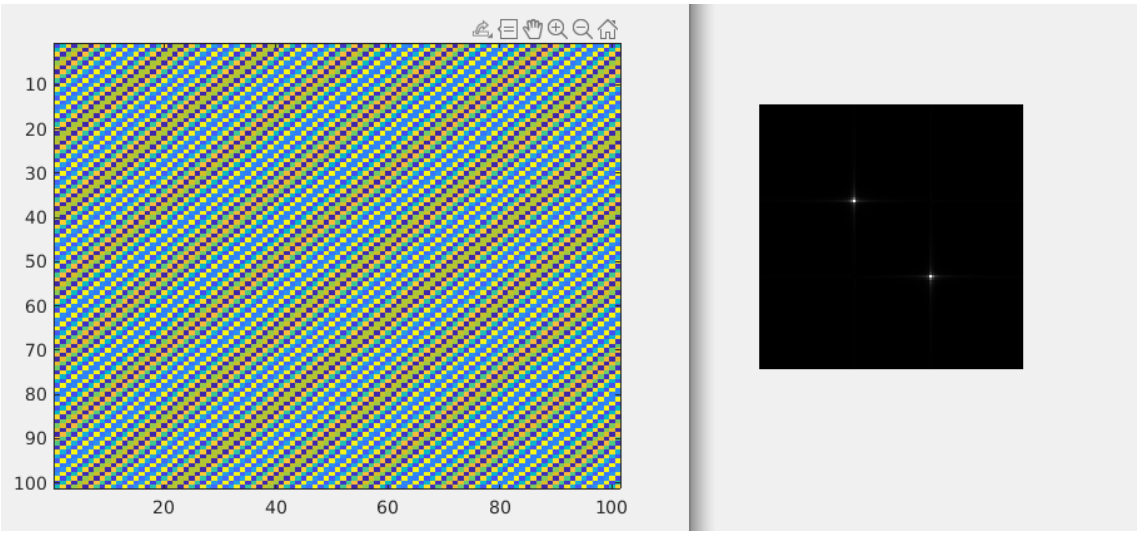
```
% Frequência entre 0 e 0.5
freq = 0.1;
```



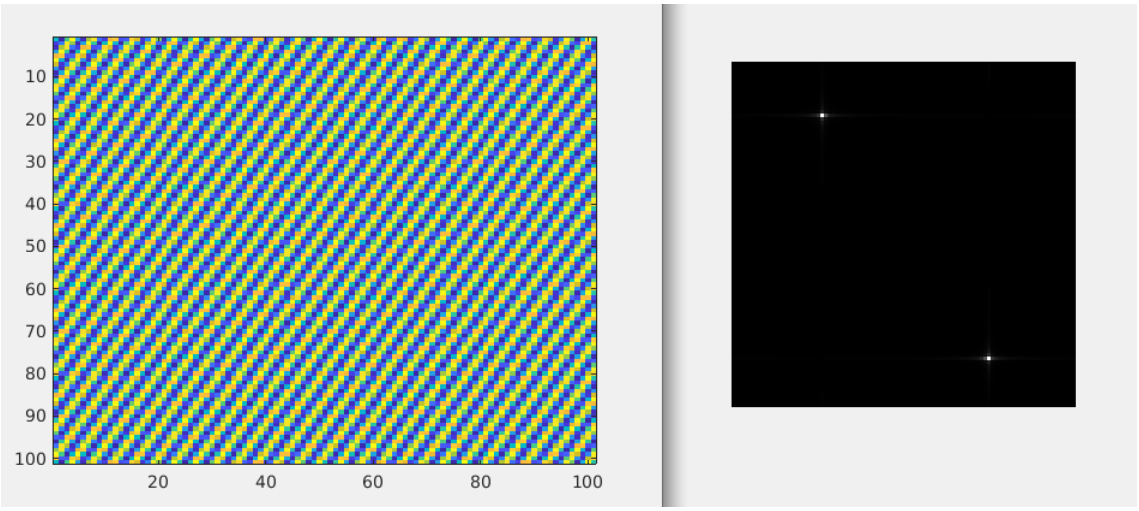

```
%  Ângulo de rotação
sinephas = pi/2;
% Frequência entre 0 e 0.5
freq = 0.1;
```




```
%  Ângulo de rotação
sinephas = pi/4;
% Frequência entre 0 e 0.5
freq = 0.5;
```

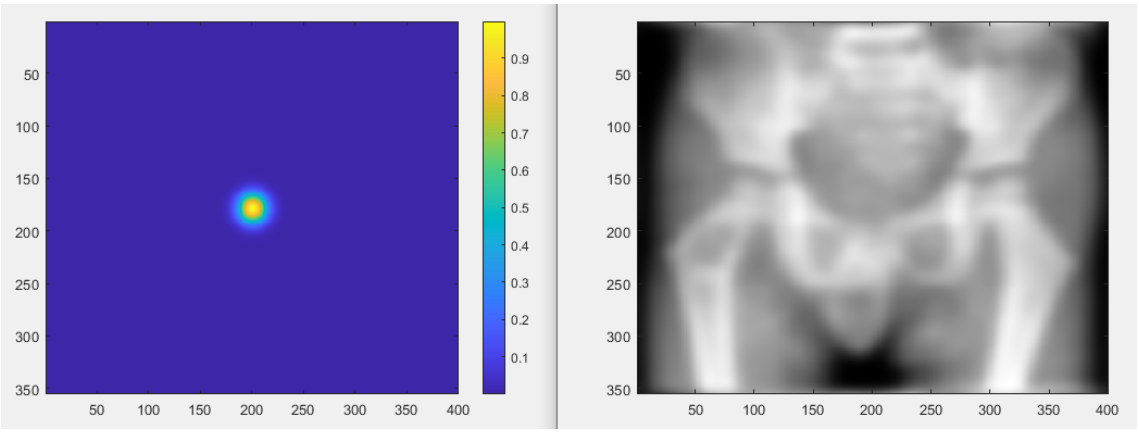



```
%  Ângulo de rotação
sinephas = pi/3;
% Frequência entre 0 e 0.5
freq = 0.3;
```

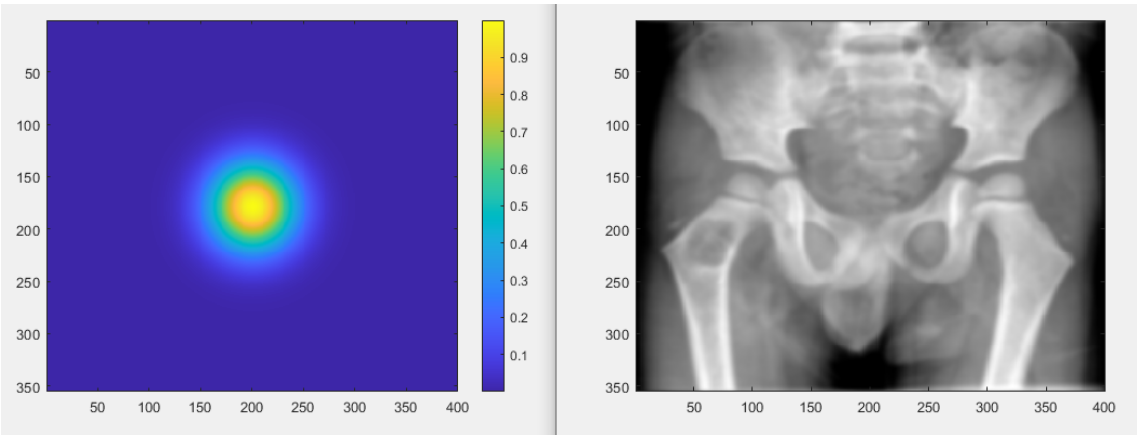


4. O código 2 colocado ao final deste roteiro pode ser usado como máscara para criar um filtro passa baixa. Varie o valor de sigma (variável: sigma = 10, 30 e 50) e comente o que acontece com o filtro e com a imagem filtrada.

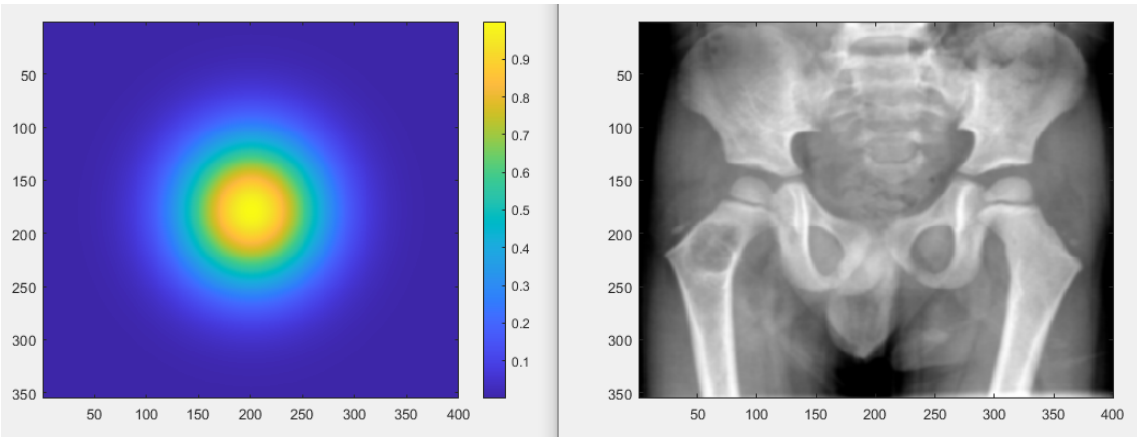
sigma = 10; % largura da gaussiana



sigma = 30; % largura da gaussiana

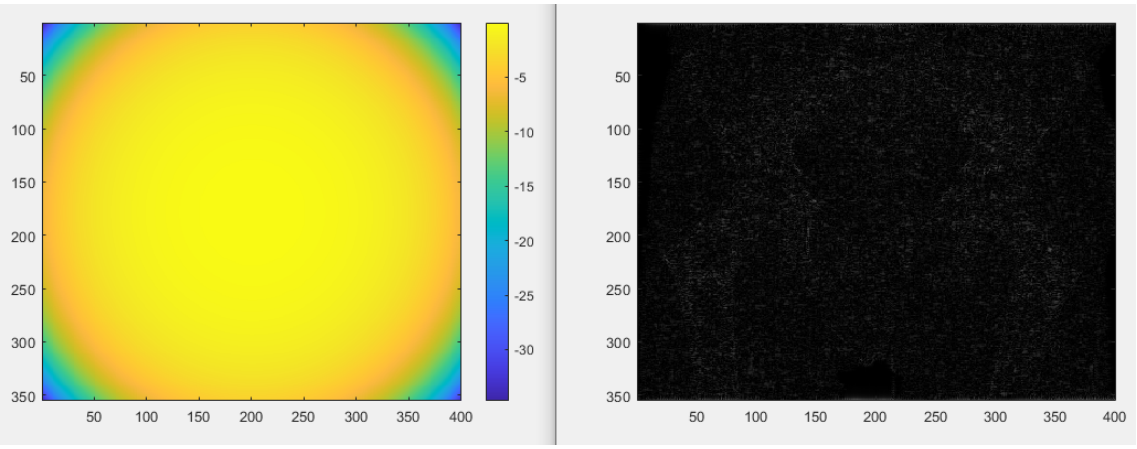


sigma = 50; % largura da gaussiana



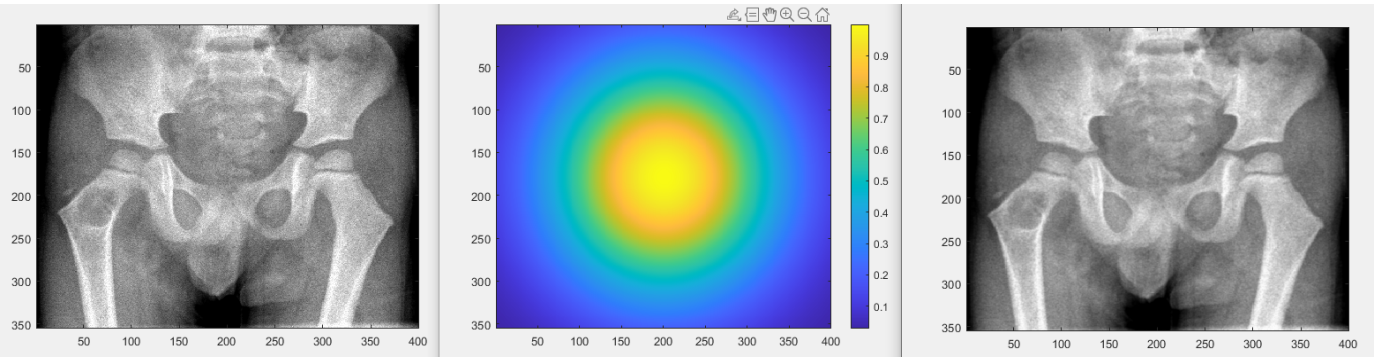
5. Crie um filtro passa alta a partir do exemplo anterior e salve como **highpassfilter.m**.

sigma = 100;



6. Crie uma rotina para ler uma imagem, adicionar ruído *poisson* e filtrar usando um filtro criado anteriormente. Salve como **NOME_p7.m**.

   sigma = 100; % largura da gaussiana



Código p7

```
%% Imagens de magnitude e fase no espaço-k

I = imread('s_PelvisRadiography.jpg');
J = rgb2gray(I);
figure(1); imagesc(J);
colormap gray;

Y = fft2(J);
```

```
Z = abs(Y);

W = fftshift(Z);
figure(2); imagesc(W);
colormap gray;

P = angle(Y);
X = fftshift(P);
figure(3); imagesc(X);
colormap gray;

%% Rotina para ler imagem, adicionar ruído poisson e
filtrar usando um filtro passa-baixa

% Filtro Passa-baixa

I = imread('s_PelvisRadiography.jpg');
ampl2 = rgb2gray(I);

limsx  = [-size(ampl2,1)/2 (size(ampl2,1)/2 -1)];
limsy  = [-size(ampl2,2)/2 (size(ampl2,2)/2 -1)];
[mx,my] = ndgrid(limsx(1):limsx(2),limsy(1):limsy(2));
sigma = 100;   % largura da gaussiana

gaus2dB = exp(-(mx.^2 + my.^2) ./ (2*sigma^2));

% imagem do filtro

figure(1);
imagesc(gaus2dB);
axis on
colorbar

% imagem normal

J = rgb2gray(I);
% figure(2); imagesc(J);
% colormap gray;

% imagem com ruído poisson

P = imnoise(J,'poisson');
figure(3); imagesc(P);
colormap gray;
```

```
% magnitude no espaço-k

Y = fft2(P);
Z1 = abs(Y);
% figure(4); imagesc(fftshift(Z1));
% colormap gray;

% aplicando o filtro no espaço-k

img = fftshift(Y) .* gaus2dB;
% figure(5); imagesc(abs(img));
% colormap gray;

% imagem normal filtrada

Z2 = ifft2(fftshift(img));
figure(6); imagesc(abs(Z2));
colormap gray;
```

Código 1

```
% especificar o vetor de frequências senoidais
sinefreq = linspace(.0001,.2,50);  % unidades arbitrárias
% linspace(X1, X2, N) gera um vetor linha de N pontos
% linearmente igualmente espaçados entre X1 e X2.%

%  Ângulo de rotação
sinephas = pi/3;
% Frequência entre 0 e 0.5
freq = 0.3;
% Iniciar a onda senoidal
lims = [-50 50];
[x,y] = ndgrid(lims(1):lims(2),lims(1):lims(2));
xp = x*cos(sinephas) + y*sin(sinephas);
% Computar o seno
img = sin(2*pi*freq*xp);
figure(1); imshow(img);

Y = fft2(img);

Z = abs(Y);
figure(2); imshow(Z,[]);
```

Código 2

```
%% Filtros no domínio da frequência

%% Filtro Passa-baixa

I = imread('s_PelvisRadiography.jpg');
ampl2 = rgb2gray(I);

limsx  = [-size(ampl2,1)/2 (size(ampl2,1)/2 -1)];
limsy  = [-size(ampl2,2)/2 (size(ampl2,2)/2 -1)];
[mx,my] = ndgrid(limsx(1):limsx(2),limsy(1):limsy(2));
sigma = 50;   % largura da gaussiana

gaus2dB = exp(-(mx.^2 + my.^2) ./ (2*sigma^2));

% imagem do filtro

figure(1);
imagesc(gaus2dB);
axis on
colorbar

%% Imagem com o filtro

% imagem normal

J = rgb2gray(I);
% figure(2); imagesc(J);
% colormap gray;

% magnitude no espaço-k

Y = fft2(J);
Z1 = abs(Y);

% figure(3); imagesc(fftshift(Z1));
% colormap gray;

% aplicando o filtro no espaço-k

img = fftshift(Y) .* gaus2dB;
```

```
% figure(4); imagesc(abs(img));
% colormap gray;

% imagem normal filtrada

Z2 = ifft2(fftshift(img));
figure(5); imagesc(abs(Z2));
colormap gray;
```

Código highpassfilter

```
%% Passa-alta

I = imread('s_PelvisRadiography.jpg');
ampl2 = rgb2gray(I);

limsx  = [-size(ampl2,1)/2 (size(ampl2,1)/2 -1)];
limsy  = [-size(ampl2,2)/2 (size(ampl2,2)/2 -1)];
[mx,my] = ndgrid(limsx(1):limsx(2),limsy(1):limsy(2));
sigma = 100;   % largura da gaussiana

gaus2dA = 1 - exp((mx.^2 + my.^2) ./ (2*sigma^2));

figure(1);
imagesc(gaus2dA);
axis on
colorbar

%% Imagem com o filtro

% imagem normal

J = rgb2gray(I);
% figure(2); imagesc(J);
% colormap gray;

% magnitude no espaço-k

Y = fft2(J);
Z1 = abs(Y);

% figure(3); imagesc(fftshift(Z1));
```

```
% colormap gray;

% aplicando o filtro no espaço-k

img = fftshift(Y) .* gaus2dA;
% figure(4); imagesc(abs(img));
% colormap gray;

% imagem normal filtrada

Z2 = ifft2(fftshift(img));
figure(5); imagesc(abs(Z2));
colormap gray;
```