Image Processing HW3 0410117 吳沛璇

**Question(1)**

**Method:**

Use fft2(X) function to do two-dimensional Fourier transform of a matrix.

fft stands for fast Fourier transform.

Because the value of mean is too large so that the whole figure is hard for observation, so I take the log of the magnitude, and it is an useful tool in image processing.

Magnitude -> log of absolute mean.

Phase -> angle of 2-dim FFT result.

**Code:**

Show Log magnitude in Figure 1.

X = imread('Fig0427(a)(woman).tif'); % 512\*512

Y = fft2(double(X)); % Compute DFT of x, 2-D transform

m = abs(Y); % Magnitude

p = unwrap(angle(Y)); % Phase

figure(1)

imshow(log(1+m),[])

title('magnitude')

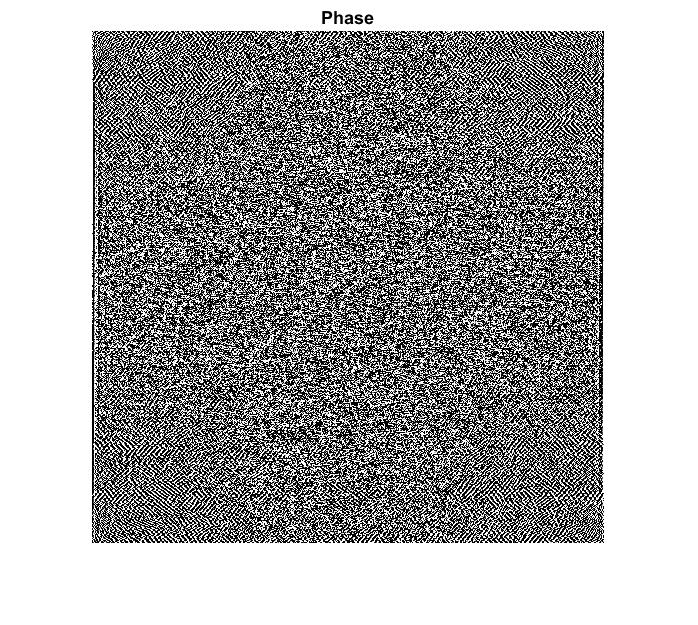
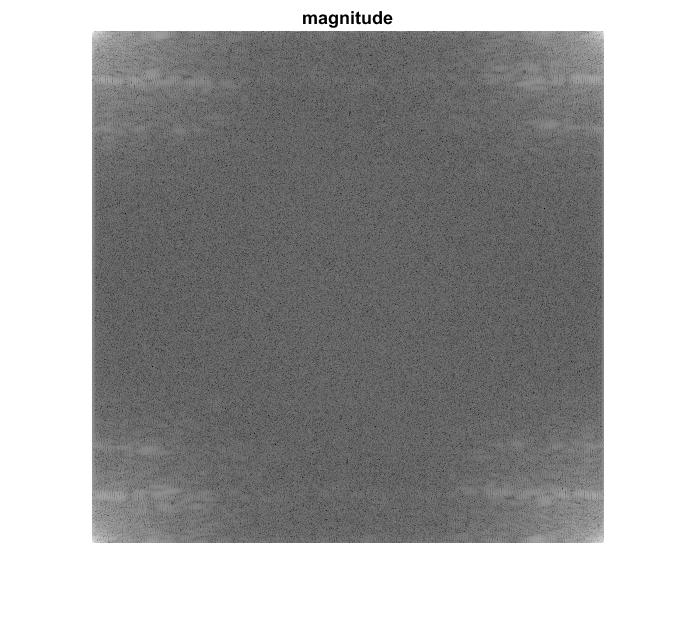
Show phase in Figure 2.

figure(2)

imshow(angle(Y))

title('Phase')

**Result:**



**Question(2)**

**Method:**

Use fftshift(X) function, it shifts zero-frequency component to center of spectrum.

Again, in order to observe the differences, magnitude is log of absolute mean.

**Code:**

Show Log magnitude in Figure 3.

imshow(log(abs(fftshift(Y)) +1),[]); % frequency scaling

title('magnitude')

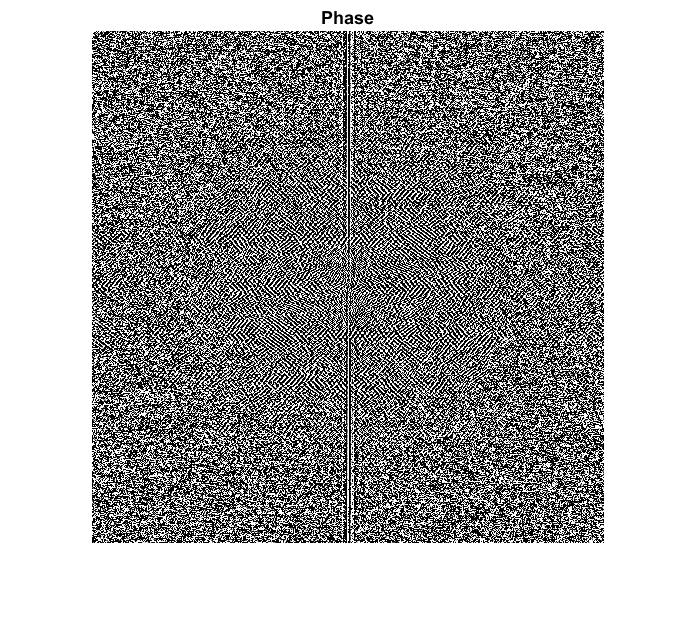
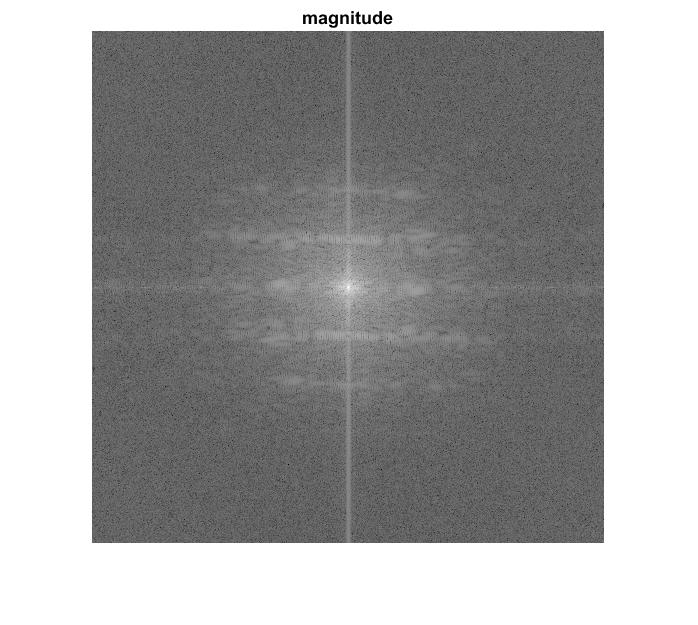
Show phase in Figure 4.

figure(4)

imshow(angle(fftshift(Y)));

title('Phase')

**Result:**



**Question(3)**

**Method:**

Use fft2(X, m, n) function to do two-dimensional Fourier transform of a matrix.

It pads X to an m\*n matrix before computing the transform.

**Code:**

%--Without Padding----------------------------------%

A = Y; % Y is the DFT of original image.

A1 = fftshift(A); % frequency scaling

[M N]=size(A); % image size

R=10; % filter size parameter

X=0:N-1;

Y=0:M-1;

[X Y]=meshgrid(X,Y);

Cx=0.5\*N; % Central of filter position in X

Cy=0.5\*M; % Central of filter position in Y

Lo=exp(-((X-Cx).^2+(Y-Cy).^2)./(2\*R).^2);

Hi=1-Lo; % High pass filter=1-low pass filter

% Filtered image=ifft(filter response\*fft(original image))

J=A1.\*Lo; % Multiply image and LPF in frequency domain

J1=ifftshift(J);

B1=ifft2(J1);

B1 = B1([1:512], [1:512]); %Because we use padding, so the filtered image will be larger. We only take original size of the picture.

% I also multiply the HPF, and observe the effect of HPF.

K=A1.\*Hi;

K1=ifftshift(K);

B2=ifft2(K1);

%----visualizing the results-----------------------------------------

figure(6)

imshow(abs(B1),[12 290]), colormap gray

title('low pass filtered image NO Padding','fontsize',14)

figure(7)

mesh(X,Y,Lo)

axis([ 0 N 0 M 0 1])

h=gca;

get(h,'FontSize')

set(h,'FontSize',14)

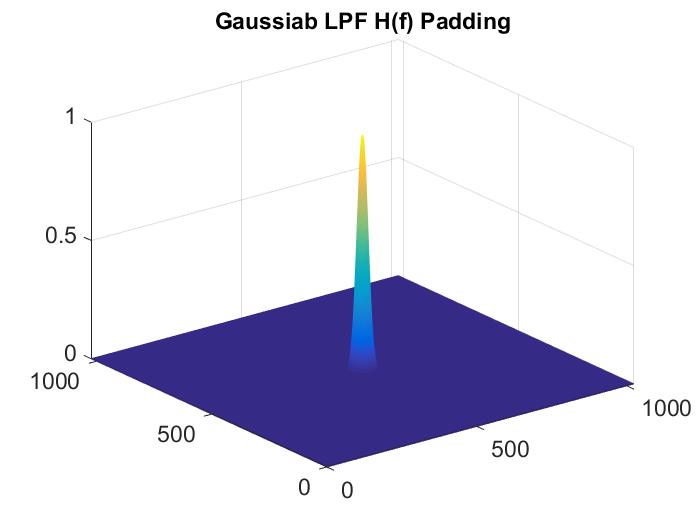
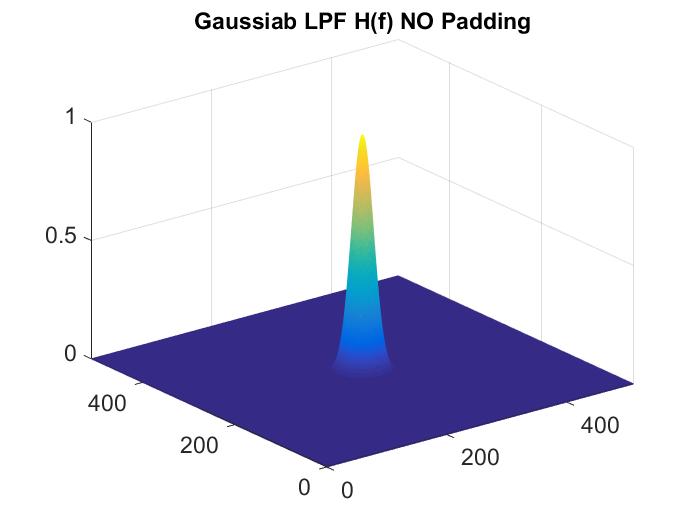
title('Gaussiab LPF H(f) NO Padding','fontsize',14)

**Result:**

Padding Image is more obscure than no-padding image.

And we can see that, the filter in padding image tends to be narrower, concentrating in the low-frequenct part. That is because the image size pad from 512\*512 to 1024\*1024, but the LFP are in fact the same.



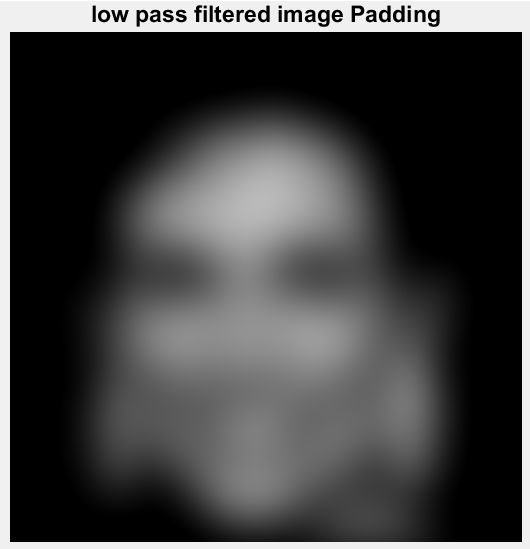
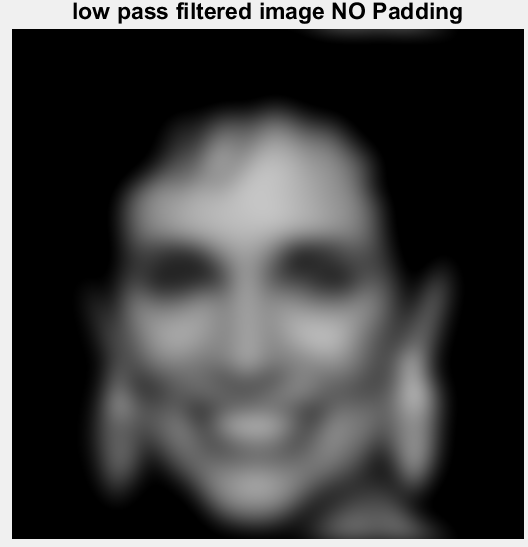


**Discussion of parameters in LPF:**

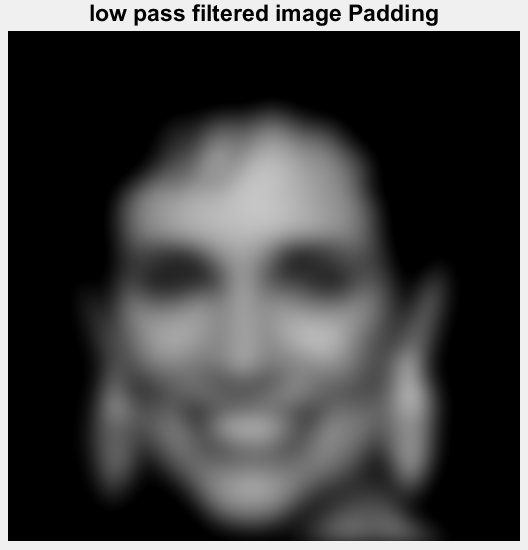
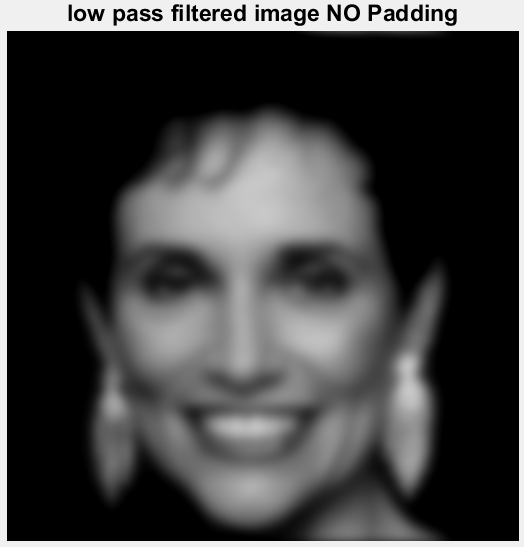
The smaller the size of filter is, the obscurer the image becomes, because LPF blocks details.

NO Padding | With Padding

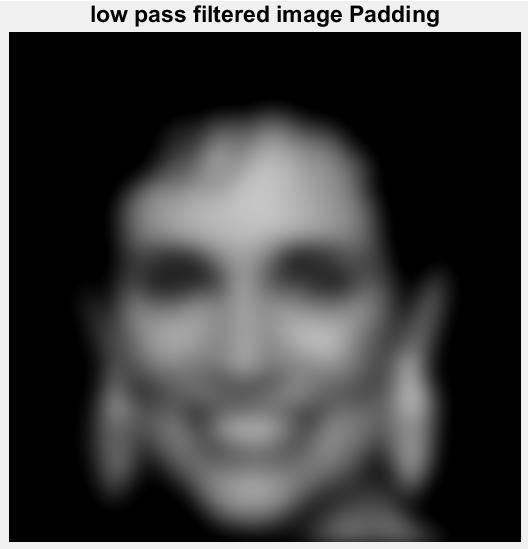
Filter radius(Size) = 5



Filter radius(Size) = 10



Filter radius(Size) = 50



Padding makes images obscure.

Because padding puts zero values in the border of an image, but they are not the original values which naturally come out from the image.