**Digital Image Processing Laboratory**

Experiment Report

Experiment Title Fourier Transform and Frequency

Domain Filtering

Student’s Name

Student’s ID

Class

Date handed in

International school, Jinan University

**A. Objectives**

1. Familiar with the concepts and principles of Fourier transform;
2. Processing digital image with MATLAB in Frequency domain;
3. Familiar with the filter design functions: fsamp2, fwind1, and fwind2.

**B. Technique**

In this project, the image *lenna.gif* will be used.

1)   Compute the forward 2D FFT of the ***lenna*** image.

2)  Lowpass Filter Design

3)  Highpass Filter Design

4)  Two Dimensional Filter Design

**C. Experiment Content**

1)   Compute the forward 2D FFT of the ***lenna*** image

a) The 2D FFT of lenna image is done by the function FFT2 in MATLAB IMAGE PROCESSING TOOLBOX. The command is the following:

imglenna=imread('lenna.gif');

imgFFT=fft2(double(imglenna)./255);

imgFFT=fftshift(imgFFT);

The function fftshift is useful for visualizing the Fourier transform with the zero-frequency (直流分量)component in the middle of the spectrum.

b)  Compute the log magnitude and phase (i.e., MATLAB IMAGE PROCESSING TOOL BOX function **ANGLE**.

The command is:

imgLogMag=log(abs(imgFFT)+1);

imgPhase=angle(imgFFT);

c)  Compute the inverse 2D FFT of the ***lenna*** image using the MATLAB IMAGE PROCESSING TOOL BOX function **IFFT2**.

The command is:

imgIFFT=abs(ifft2(imgFFT))

d)  Plot the original ***lenna*** image, log magnitude, phase, and inverse transformed images.

The command is:

figure;

subplot(221);

imshow(imglenna);

title('Original Image');

subplot(222);

imshow(imgLogMag,[]);

title('Log Maganitude of FFT');

subplot(223);

imshow(imgPhase,[]);

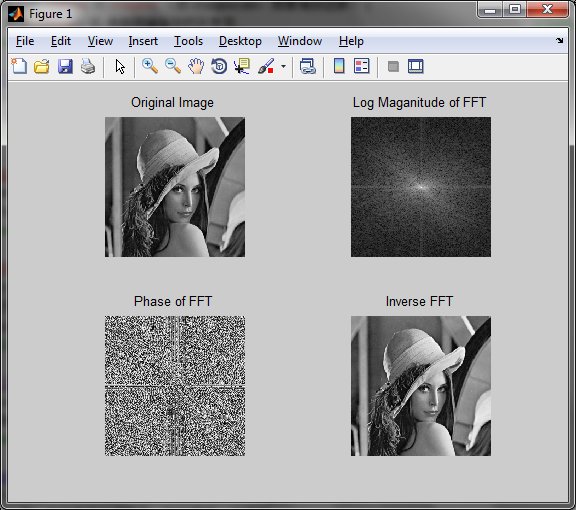
title('Phase of FFT');

subplot(224);

imshow(imgIFFT,[]);

title('Inverse FFT');

The result is:



***Figure 1:*** *Original, log magnitude, phase, and inverse transformed images*

2)  Lowpass Filter Design

a) Use the MATLAB IMAGE PROCESSING TOOL BOX function FSPECIAL to design an 11x11 Gaussian lowpass filter with a value of s equal to 1.3.

The command is:

LowpassFilter=fspecial('gaussian',[11 11],1.3);

b) Compute the forward 2D FFT of the filter kernel using the same size FFT as that of the lenna image. Utilize the SIZE function from the example on the website.

The command is:

imgSize=size(imglenna);

imgRows=imgSize(1);

imgCols=imgSize(2);

LowpassFFT=fftshift(fft2(LowpassFilter,imgRows,imgCols));

We can get the row and column of the image and use the function fft2 to Compute the forward 2D FFT of the filter kernel

c) From the results in b) compute and plot the log magnitude and phase of the Gaussian Lowpass Filter kernel.

The command is:

figure;

subplot(121);

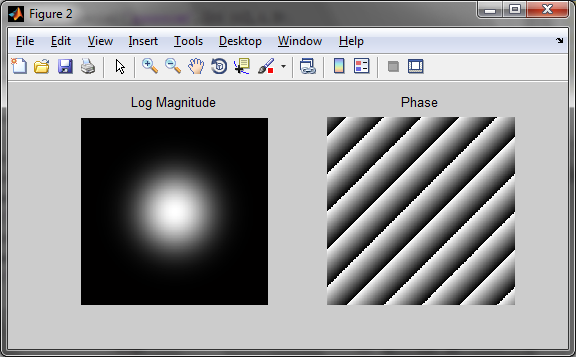
imshow(log(abs(LowpassFFT)+1),[]);

title('Log Magnitude');

subplot(122);

imshow(angle(LowpassFFT),[]);

title('Phase');



***Figure 2:*** *log magnitude and phase of the Gaussian Lowpass Filter kernel*

From the result, we can find the center of log magnitude is bright and the phase is alternating light and dark stripes.

d) Utilizing the results in 1) and 2b) perform the filtering function G(u,v)=H(u,v)\*F(u,v), where H(u,v)=2D FFT of the Gaussian Filter Kernel, and F(u,v)=2D FFT of the lenna image. Plot the log magnitude and phase of the lowpass filtered image.

The command is:

imgFiltered=LowpassFFT .\* imgFFT;

figure;

subplot(121);

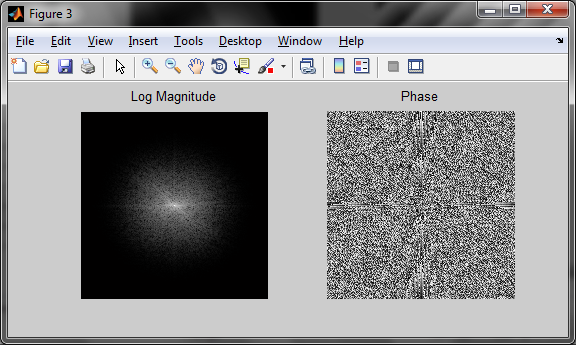
imshow(log(abs(imgFiltered)+1),[]);

title('Log Magnitude');

subplot(122);

imshow(angle(imgFiltered),[]);

title('Phase');



***Figure 3:*** *log magnitude and phase of the lowpass filtered image*

e) Compute and plot the inverse 2D FFT of the lowpass filtered image.

The command is:

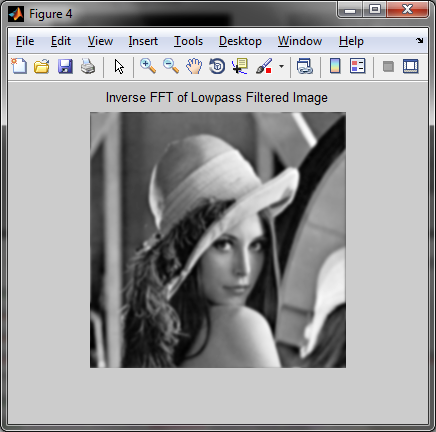
figure;

imgLowpassFiltered=abs(ifft2(imgFiltered));

imgLowpassFiltered=circshift(imgLowpassFiltered,[-1.\*floor(length(LowpassFilter)/2) -1.\*floor(length(LowpassFilter)/2)]);

imshow(imgLowpassFiltered,[]);

title('Inverse FFT of Lowpass Filtered Image');



***Figure 4:*** *the inverse 2D FFT of the lowpass filtered image*

From the result, we can find the image after lowpass filiter is fuzzy and the high-frequency components of the image are lost.

3)  Highpass Filter Design

a) Use the MATLAB IMAGE PROCESSING TOOL BOX function FSPECIAL to design a laplacian highpass filter.

The command is:

HighpassFilter=fspecial('laplacian');

b) Compute the forward 2D FFT of the filter kernel using the same size FFT as that of the lenna image. Utilize the SIZE function from the example on the website.

The command is:

HighpassFFT=fftshift(fft2(HighpassFilter,imgRows,imgCols));

c) From the results in b) compute and plot the log magnitude and phase of the Laplacian highpass Filter kernel.

The command is:

figure;

subplot(121);

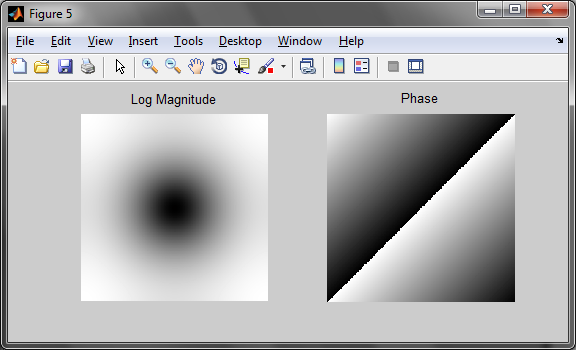
imshow(log(abs(HighpassFFT)+1),[]);

title('Log Magnitude');

subplot(122);

imshow(angle(HighpassFFT),[]);

title('Phase');



***Figure 5:*** *log magnitude and phase of the Laplacian highpass Filter kernel*

From the result, we can find the center of the log magnitude is dark and there is a clear dividing line in phase.

d) Utilizing the results in 1) and 3b) perform the filtering function G(u,v)=H(u,v)\*F(u,v), where H(u,v)=2D FFT of the Gaussian Filter Kernel, and F(u,v)=2D FFT of the lenna image. Plot the log magnitude and phase of the lowpass filtered image.The command is:

imgFiltered=HighpassFFT .\* imgFFT;

figure;

subplot(121);

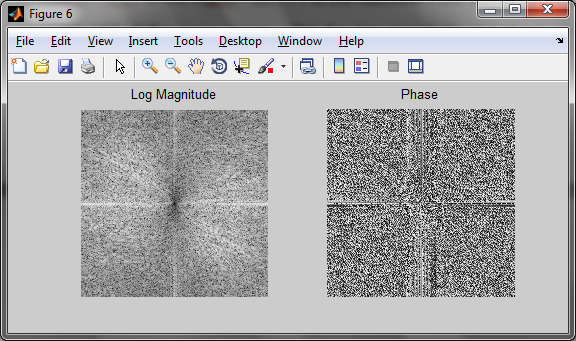
imshow(log(abs(imgFiltered)+1),[]);

title('Log Magnitude');

subplot(122);

imshow(angle(imgFiltered),[]);

title('Phase');



***Figure 6:*** *log magnitude and phase of the lowpass filtered image*

e) Compute and plot the inverse 2D FFT of the highpass filtered image using the IFFT2 function.

The command is:

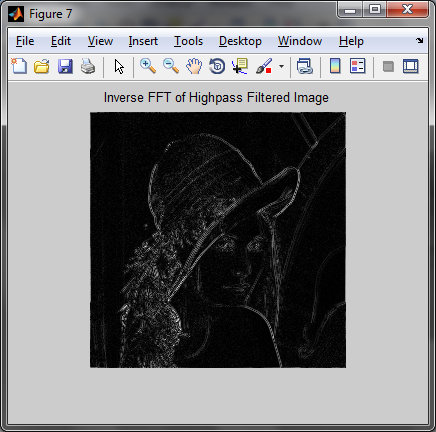
figure;

imgHighpassFiltered=abs(ifft2(imgFiltered));

imgHighpassFiltered=circshift(imgHighpassFiltered,[-1.\*floor(length(HighpassFilter)/2) -1.\*floor(length(HighpassFilter)/2)]);

imshow(imgHighpassFiltered,[]);

title('Inverse FFT of Highpass Filtered Image');



***Figure 7:*** *Inverse FFT of Highpass Filtered Image*

From the result, we can find the image after Highpass Filter only contains the contour with frequency components.

4)  Two Dimensional Filter Design

a)  The objective of this exercise id to utilize the filter design functions: fsamp2, fwind1, and fwind2.

1. Use [f1,f2]=freqspace(21,'meshgrid'); command to design the sampling grid for the filter.

2.  Once 1) is completed compute the radius vectors for the following filter designs for the filter design functions: fsamp2, fwind1, and fwind2.

  a)  The radius vectors are the following:

Bandpass: (r<0.1) | (r>0.6)

Lowpass:  r>0.6

Highpass:  r<0.6

The command is:

[f1,f2]=freqspace(21,'meshgrid');

r=sqrt(f1.^2 + f2.^2);

Hd=ones(size(f1));

Bandpass=Hd;

Lowpass=Hd;

Highpass=Hd;

Bandpass((r<0.1)|(r>0.6))=0;

Lowpass(r>0.6)=0;

Highpass(r<0.6)=0;

 b)  For each of the filtering algorithms do the following:

1)  Design a bandpass, lowpass, and highpass filter

2)  Compute the forward 2D FFT of the filter kernels using the same size FFT as that of the lenna image.  Utilize the SIZE function from the example on the website.

3)  Use the results in 2) compute and plot the log magnitude and phase of each respective filter kernel.

4)  Utilizing the results in 2) perform the filtering function G(u,v)=H(u,v)\*F(u,v), where H(u,v)=2D FFT of the respective filter kernel, and F(u,v)=2D FFT of the lenna image.  Plot the log magnitude and phase of the filtered image.

5)  Compute and plot the inverse 2D FFT of each filtered image.

With the filter design functions fsamp2, the command is:

BandpassFilter=fsamp2(Bandpass);

LowpassFilter=fsamp2(Lowpass);

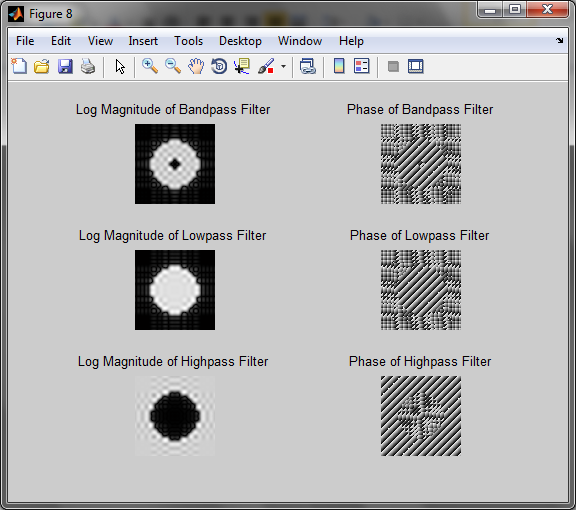
HighpassFilter=fsamp2(Highpass);

BandpassFilterFFT=fftshift(fft2(BandpassFilter,imgRows,imgCols));

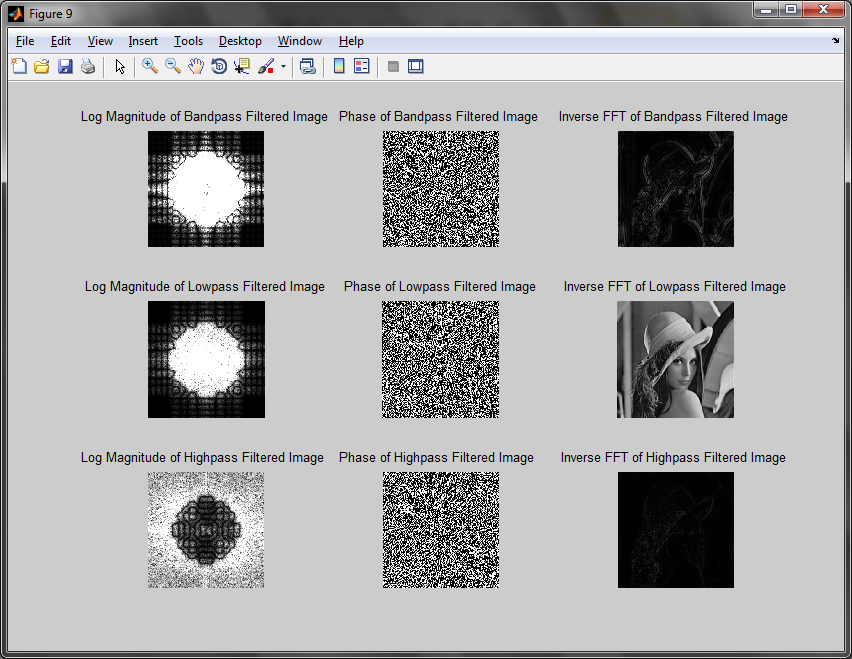
LowpassFilterFFT=fftshift(fft2(LowpassFilter,imgRows,imgCols));

HighpassFilterFFT=fftshift(fft2(HighpassFilter,imgRows,imgCols));

The results are:



***Figure 8:****Log Magnitude and Phase of bandpass, lowpass, and highpass filter*



***Figure 9:*** *log magnitude, phase and inverse 2D FFT of each filtered image*

From the results, we can find that the image after Lowpass Filter is fuzzy. The image after Highpass and Bandpass is distortion but retain different frequency components of the image.

With the filter design functions fwind1, the command is:

BandpassFilter=fwind1(Bandpass, hamming(21));

HighpassFilter=fwind1(Highpass, hamming(21));

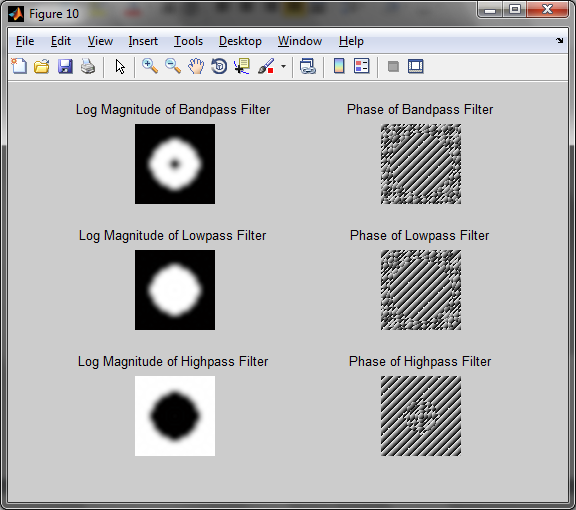
LowpassFilter=fwind1(Lowpass, hamming(21));

BandpassFilterFFT=fftshift(fft2(BandpassFilter,imgRows,imgCols));

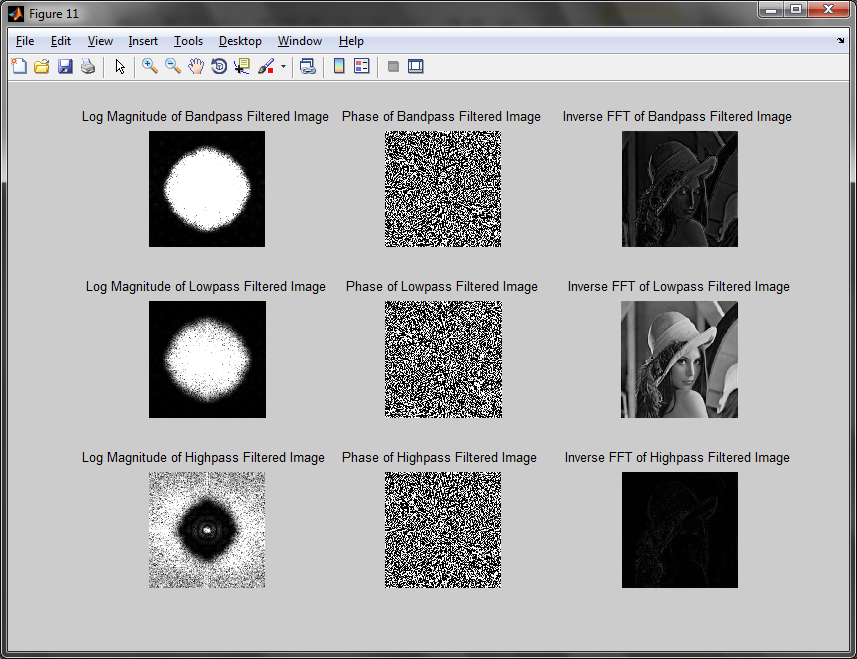
LowpassFilterFFT=fftshift(fft2(LowpassFilter,imgRows,imgCols));

HighpassFilterFFT=fftshift(fft2(HighpassFilter,imgRows,imgCols));

The results are:



***Figure 10:*** *Log Magnitude and Phase of bandpass, lowpass, and highpass filter*



***Figure 11:*** *log magnitude, phase and inverse 2D FFT of each filtered image*

From the results, we can find that image after Lowpass Filter is better. The image after Highpass Filter retains minimal information. The image after Bandpass Filter retain part of information.

With the filter design functions fwind2, the command is:

window=fspecial('gaussian', 21, 2);

window=window ./ max(max(window));

BandpassFilter=fwind2(Bandpass, window);

HighpassFilter=fwind2(Highpass, window);

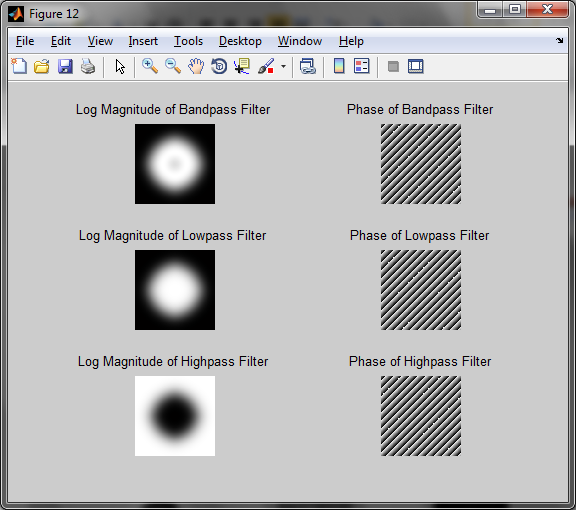
LowpassFilter=fwind2(Lowpass, window);

BandpassFilterFFT=fftshift(fft2(BandpassFilter,imgRows,imgCols));

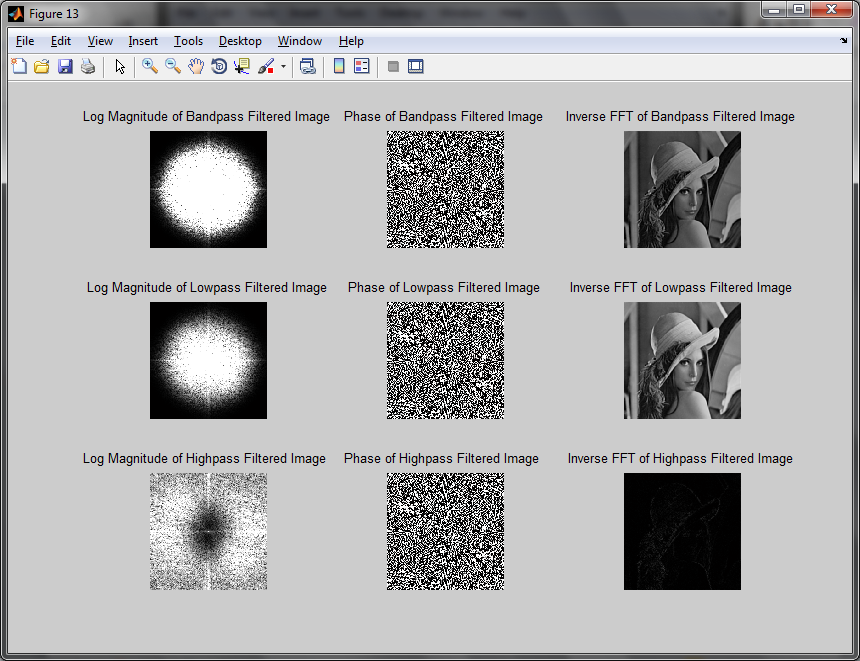
LowpassFilterFFT=fftshift(fft2(LowpassFilter,imgRows,imgCols));

HighpassFilterFFT=fftshift(fft2(HighpassFilter,imgRows,imgCols));

The results are:



***Figure 12:*** *Log Magnitude and Phase of bandpass, lowpass, and highpass filter*



***Figure 13:*** *log magnitude, phase and inverse 2D FFT of each filtered image*

From the results, we can find that the image after Highpass Filter retain high frequency components of the image contour. The image after Lowpass Filter lost less information compared with the original image. And the image after Bandpass Filter is dark compared with the original image.

**D. Conclusions**

The effect of Lowpass Filter in Frequency domain is to remove the high frequency noise of the image, and the ability depends on the form and cutoff frequency of the filter. Lowpass Filter will also cause image fuzzy by removing the noise.

Highpass Filter can enhance the image edge, which is the basis of image edge detection. Highpass Filter basically suppresses the image of smooth information.

Bandpass Filter performs better to remove particular type noise. The gray level of the image is obviously compression after Fourier inverter.