电子科技大学

University of Electronic Science and Technology of China

实验报告

实验课程： 光电图像处理

姓　　名： 李 宁

学　　号： 2016050201017

实验地点： 211 楼 909

指导老师： 张 静

实验时间： 2018 年 10 月 17 日

一、实验名称：数字图像的 DFT/DCT 及频域滤波

二、实验目的

1、熟练掌握数字信号（1D）及数字图像(2D)离散傅立叶变换（DFT）方法、基本原理及实现流程。

2、深入理解离散信号采样频率、奈奎斯特频率及频率分辨率等基本概念，弄清它们之间的相互关系。了解离散傅里叶变换（DFT）中频率泄露的原因，以及如何尽量减少频率泄露影响的途径。

3、熟悉和掌握利用 MATLAB 工具进行 1D/2DFFT 的基本步骤、MATLAB 函数使用及对具体变换的处理流程。

4、能熟练应用 MATLAB 工具对数字图像进行 FFT 及 DCT 处理，并能根据需要进行必要的频谱分析和可视化显示。

5、熟悉和掌握几种典型的频域低通滤波器及高通滤波器的原理、特性和作用。

6、搞清空域图像处理与频域图像处理的异同，包括处理流程、各自的优势等。掌握频域滤波的基本原理和基本流程，并能编写出相应的程序代码。

三、实验原理

1、实际信号的频谱分析及频域滤波

1)、利用傅里叶变换对信号进行频域分析。

正变化：

$$F(u)=\int_{-\infty}^{+\infty} f(t)e^{-j2\pi ut}dt$$

逆变换：

$$f(t)=\int_{-\infty}^{+\infty} F(u)e^{j2\pi ut}du$$

二维离散正变换：

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

$$u=0,1,2,...,M-1,\ \ v=0,1,2,...,N-1$$

二维离散逆变换：

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

$$x=0,1,2,...,M-1,\ \ y=0,1,2,...,N-1$$

2)、利用巴特沃斯滤波器对图像进行滤波处理。

BLPF 传递函数：

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

$$D(u,v)=\sqrt{(u-P/2)^2+(v-Q/2)^2}$$

这里 D0 为截止频率，n 为阶数。当 D(u,v)=D0 时，H(u,v)降为最大值的 50% 。P、Q 意义如前所述。

2、验证空频域滤波结果的一致性

1)、频谱中心化

图像为了分析傅立叶频谱,常将傅氏频谱原点移到矩阵 M×N 中心。(频谱中心化)

图像 f(x,y) 乘上(-1)x+y 因子后进行 DFT 变换

$$\begin{cases}u_0=M/2\\ v_0=N/2\end{cases},\quad e^{j2\pi(u_0x/M+v_0y/N)}=e^{j\pi(x+y)}=(-1)^{x+y}$$

$$f(x,y)(-1)^{(x+y)}\Leftrightarrow F(u-M/2,v-N/2)$$

空域 f(x,y)经过变换后的 F(u,v)是以坐标原点中心为对称的。

F(0, 0)置于谱方阵的中心，其余各行各列的谱对中心都是共轭对称的。

四、实验步骤

1、实际信号的频谱分析及频域滤波

1)、从保存在本地磁盘的文本文件中读入一实际数字信号，该磁盘文件名为:“seismic_251_301_2ms.txt”，已知该信号的时间采样率为 dt=2ms。文件中的信号由 301 个等长的按列排列的一维列信号组成，每个一维列信号有 251 个采样点，信号实际计时起点为 1800ms，延时长度为 L=(251-1)*2ms=500ms。请读出其中的某一列信号，并画出该信号振幅随时间变化的波形图，以 ms 为时间单位。

2)、对第一步中抽取的其中一列信号做快速傅里叶变换（FFT），分别画出频谱中心化的对称频谱和只含有正半抽的信号频谱图，并对该信号做简要的频谱分析。要求规范的标注纵横坐标实际物理量和对应的单位。

3)、设定截止频率 D0=100，试在同一张图上以不同线型画出 n =1，2，4 阶下的巴特沃思（Buttworth）低通滤波器（一维）的频率响应曲线。要求标注规范地纵横坐标实际物理量和对应的单位。

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

其中，D0 为截止频率，N 为滤波器长度，n 为滤波器阶数。

4)、选择合适的 D0，利用上述 2 阶 Butterworth 低通滤波器，对第 1）步读取的列信号进行滤波实验。并分析截止频率对滤波效果的影响。

2、验证空频域滤波结果的一致性

1)、任意读取一幅 8bit 灰度图像 f，给图像加入均值为 0，方差 0.02 的高斯噪声。

2)、利用 9×9，标准差为 2 的空域高斯滤波器 h 对加噪声图像 f 进行空域滤波。滤波中，要求以重复像素方式处理边界问题。

3)、利用第 2）步产生的滤波器 h，编程计算其对应的频域滤波器 H（考虑填充滤波效果，尺寸由输入的待滤波图像 f 决定）。

4）对加噪声图像 f 进行频域滤波。并把滤波结果与空域滤波结果进行对比，检验两种结果的一致性。

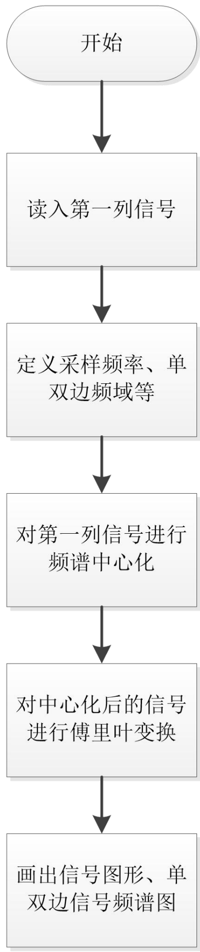
5）分别画出原始图像和加噪图像的中心化频谱图，空域 h 平面图，空域滤波结果及频谱图，中心化频域 H 平面图和 3D 图，频域滤波结果及其频谱（中心化）图等。

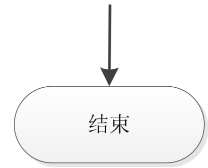
五、实验结果及分析

1、实际信号的频谱分析及频域滤波

1)、读取的为第一列信号，第一张为波形图，第二张为频谱中心化的对称频谱，第三章为只含有正半轴的对称频谱。

流程图：





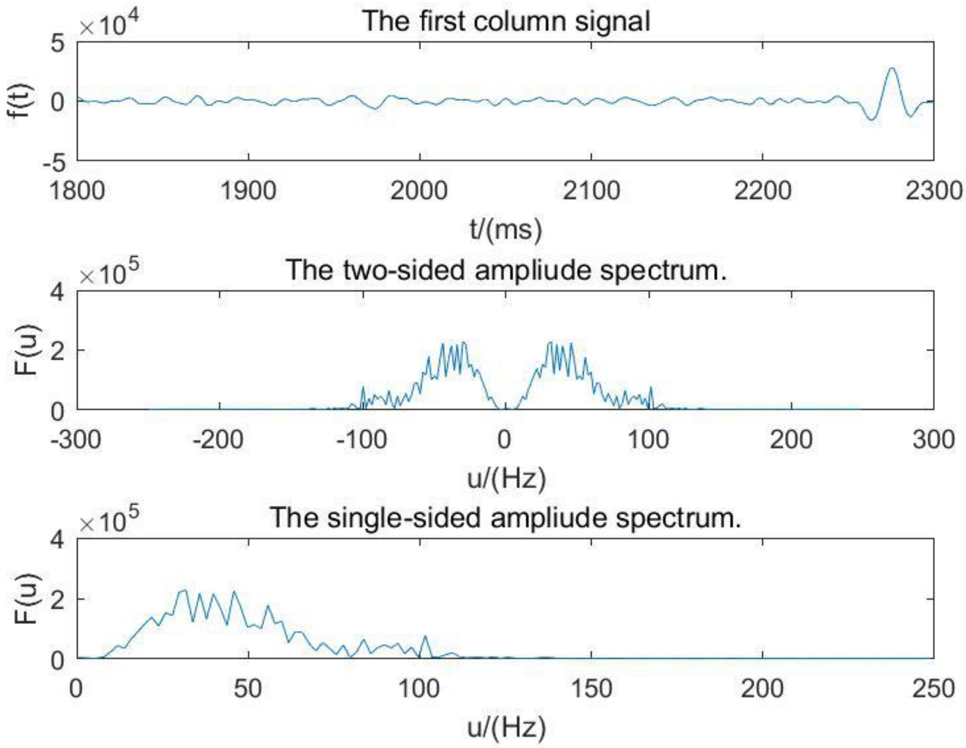
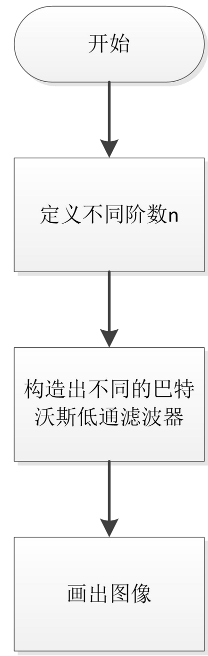
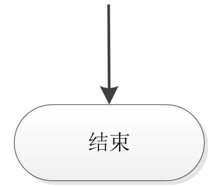


图 1 波形图

2）、n=1,2,4 阶下的 Butterworth 低通滤波器的频响曲线：

流程图：





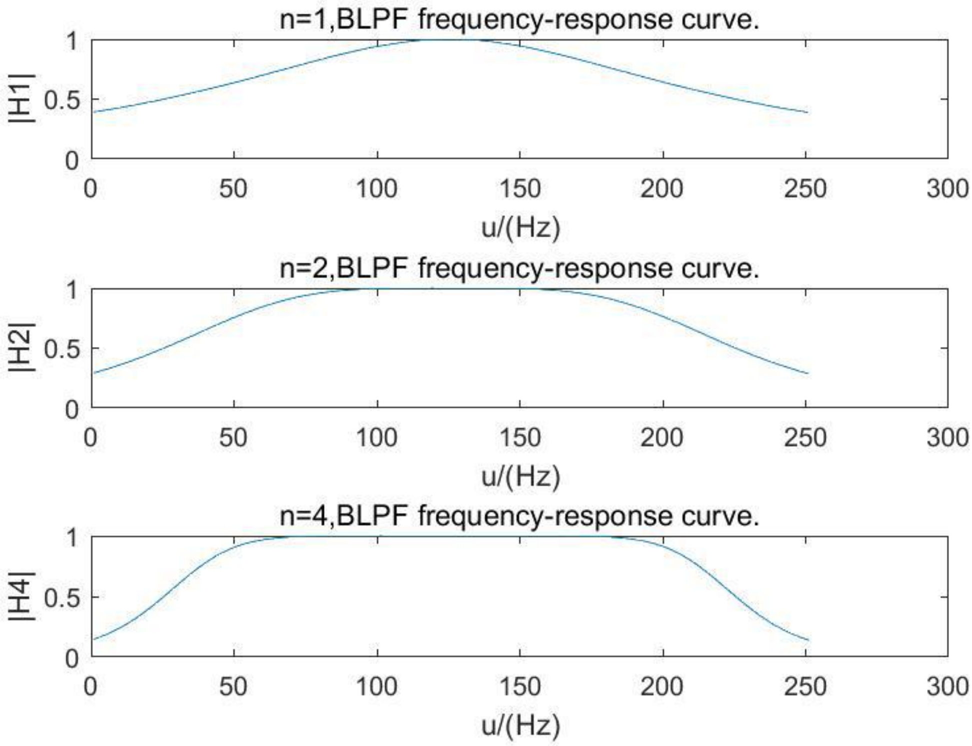
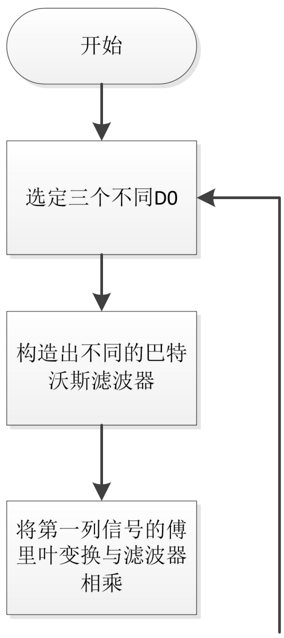
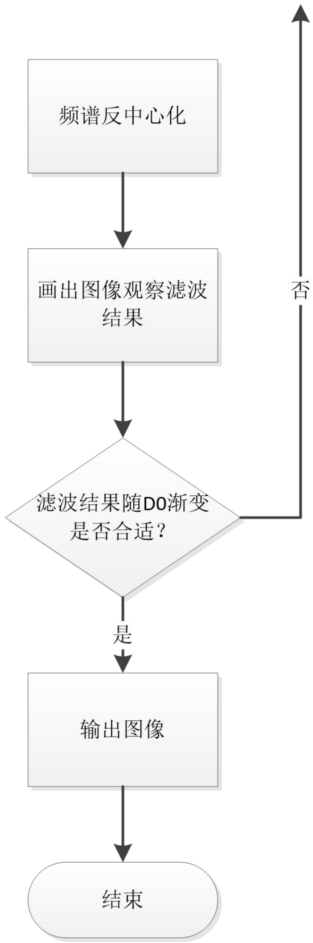


图 2 n=1,2,4 时，|H|

3）、D0=5、20、50，




频谱反中心化
画出图像观察滤波结果
滤波结果随D0渐变是否合适？
否
是
输出图像
结束

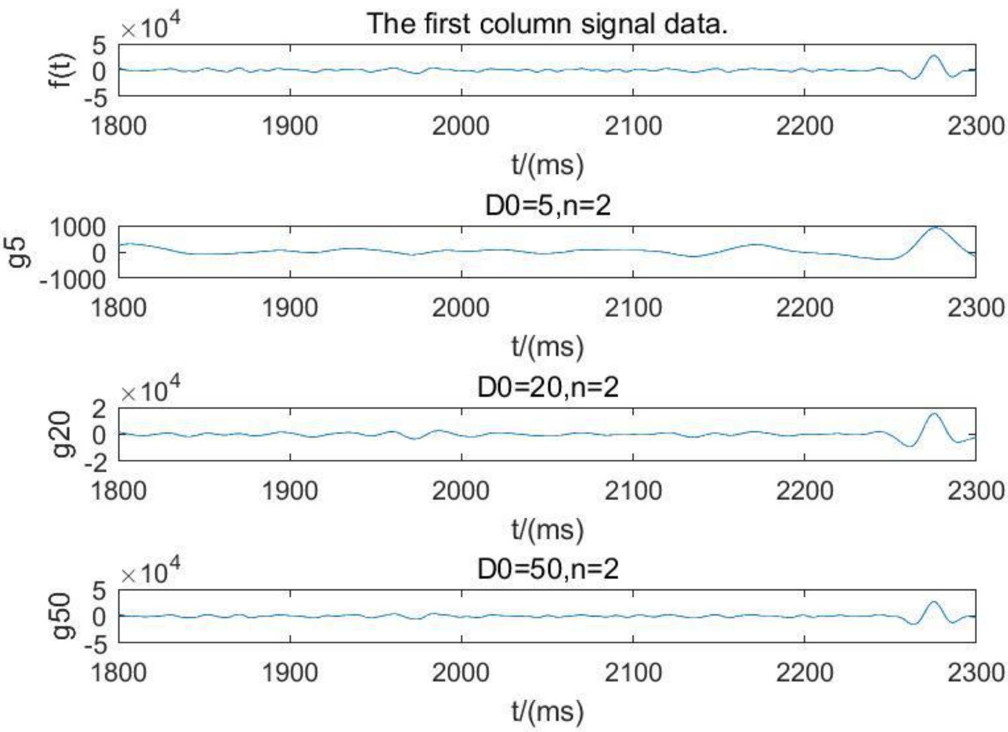

The first column signal data.
f(t)
$\times 10^4$
5
0
-5
1800
1900
2000
2100
2200
2300
t/(ms)
D0=5,n=2
g5
1000
0
-1000
t/(ms)
D0=20,n=2
g20
$\times 10^4$
2
0
-2
t/(ms)
D0=50,n=2
g50
$\times 10^4$
5
0
-5
t/(ms)

图 3 D0=5、20、50

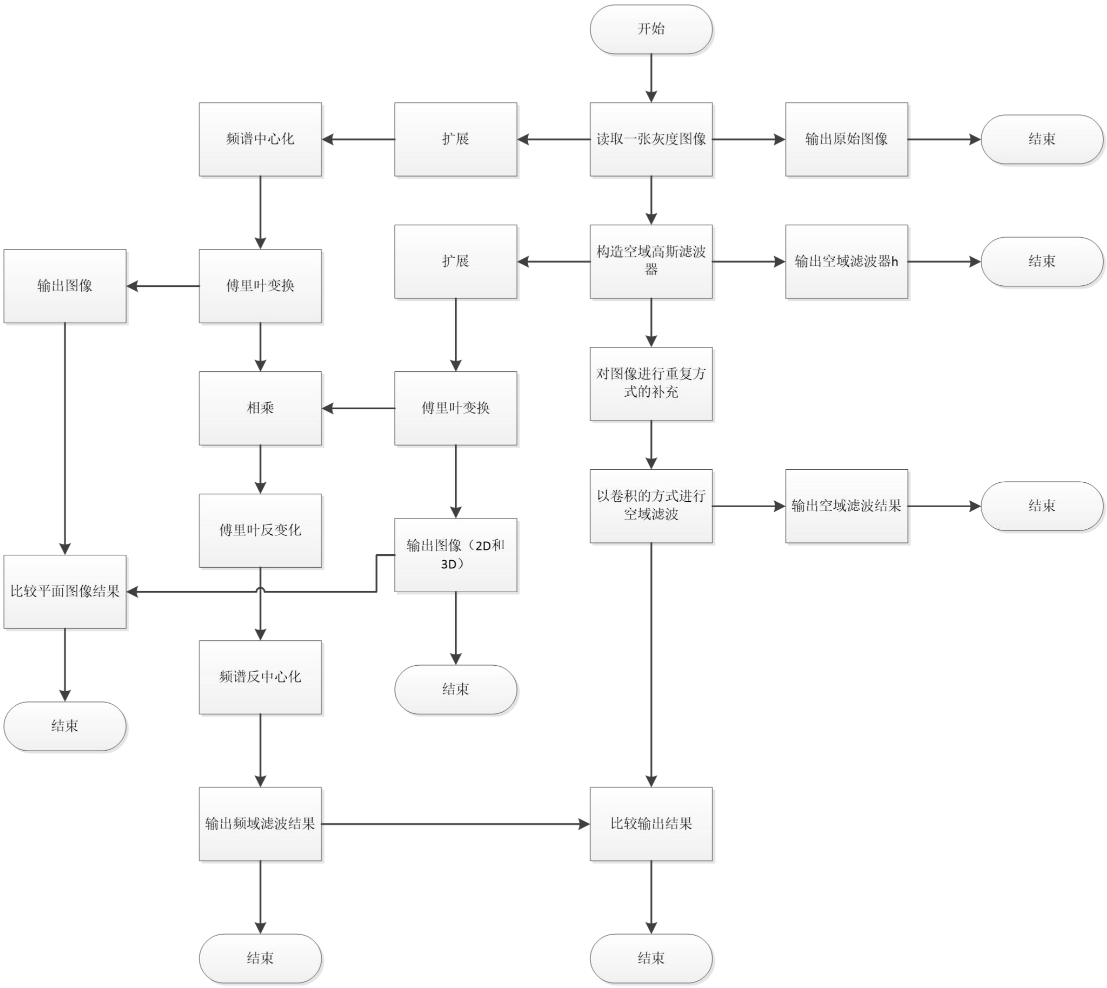
截止频率对滤波效果的影响：

随着截止频率 D0 的减小，信号曲线愈发平滑，滤波效果越好。

2、验证空频域滤波结果的一致性

1）、原始图像、加噪声之后的图像、空域滤波后的图像、频域滤波后的图像。

流程图：



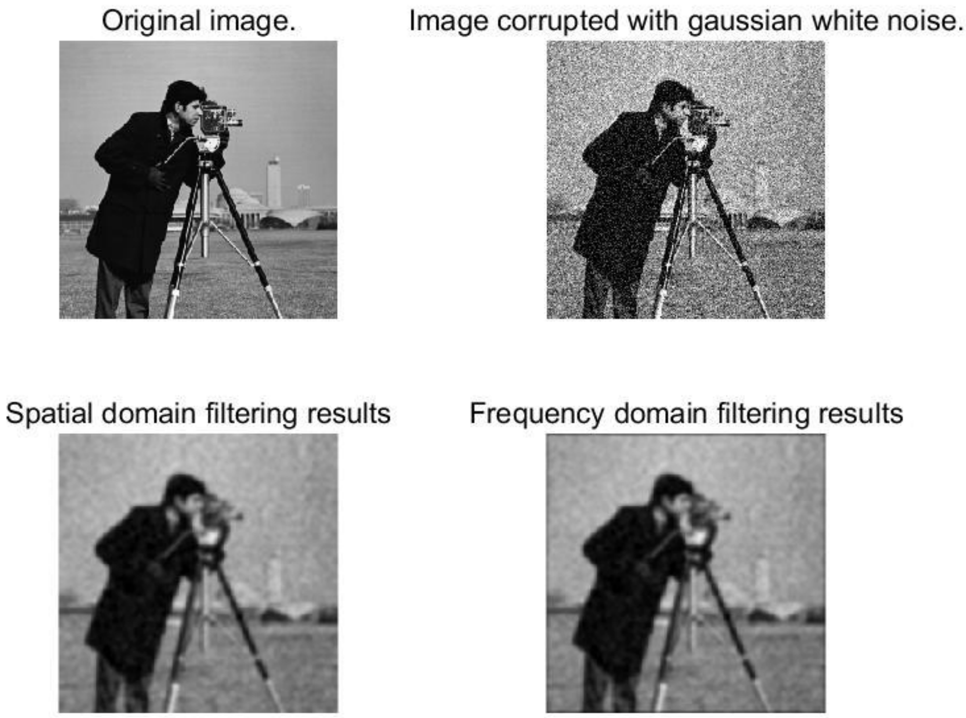


图 4 图像对比

由滤波结果我们可以看出空域滤波和频域滤波的最终图像结果无明显差别，但是在实验中我发现频域滤波结果的四周有黑边现象。

2）、原始图像、加高斯噪声后、空域滤波后、频域滤波后的中心化频谱图像。

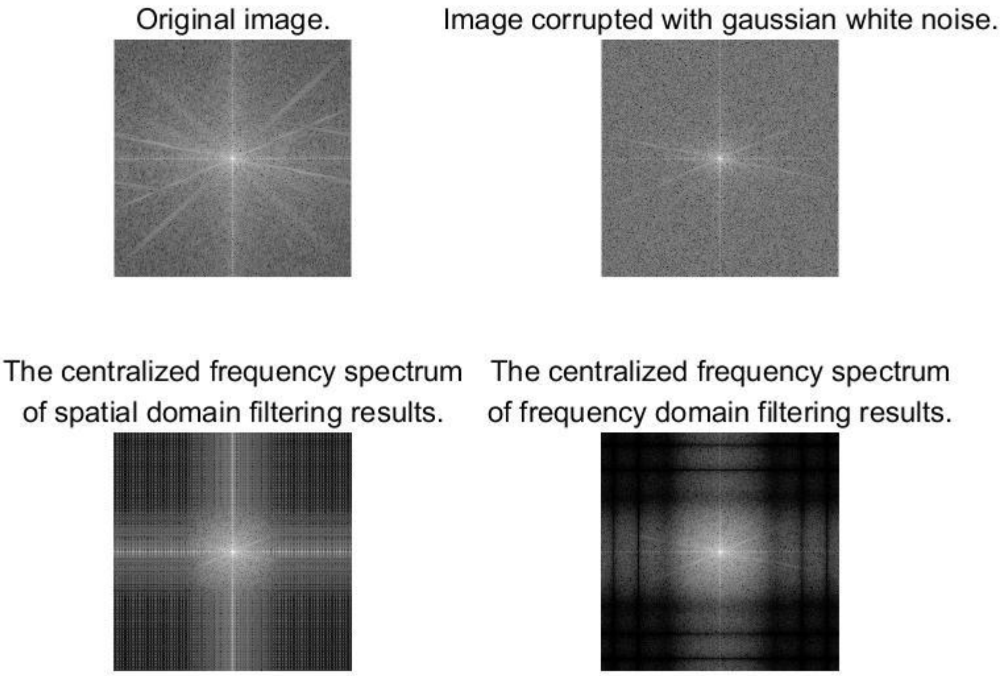


图 5 中心化频谱对比

我们看出空域滤波和频率滤波结果的中心化频谱图有差异。

3)、空域滤波器 h 平面图、空域滤波结果、中心化频谱图。

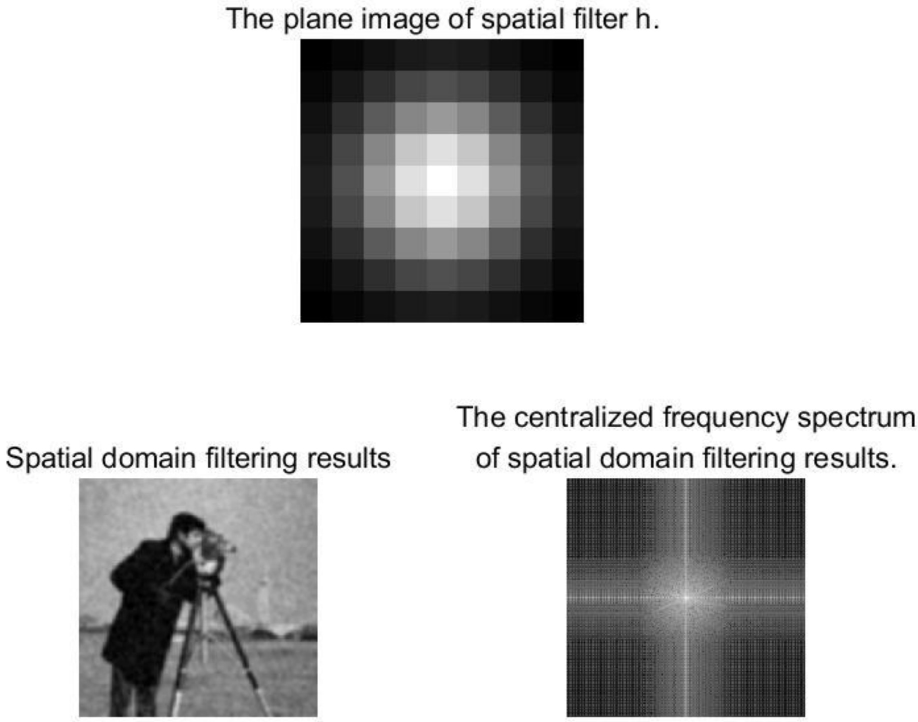


图 6 空域滤波

4）、中心化频域滤波器 H 3D 图像。

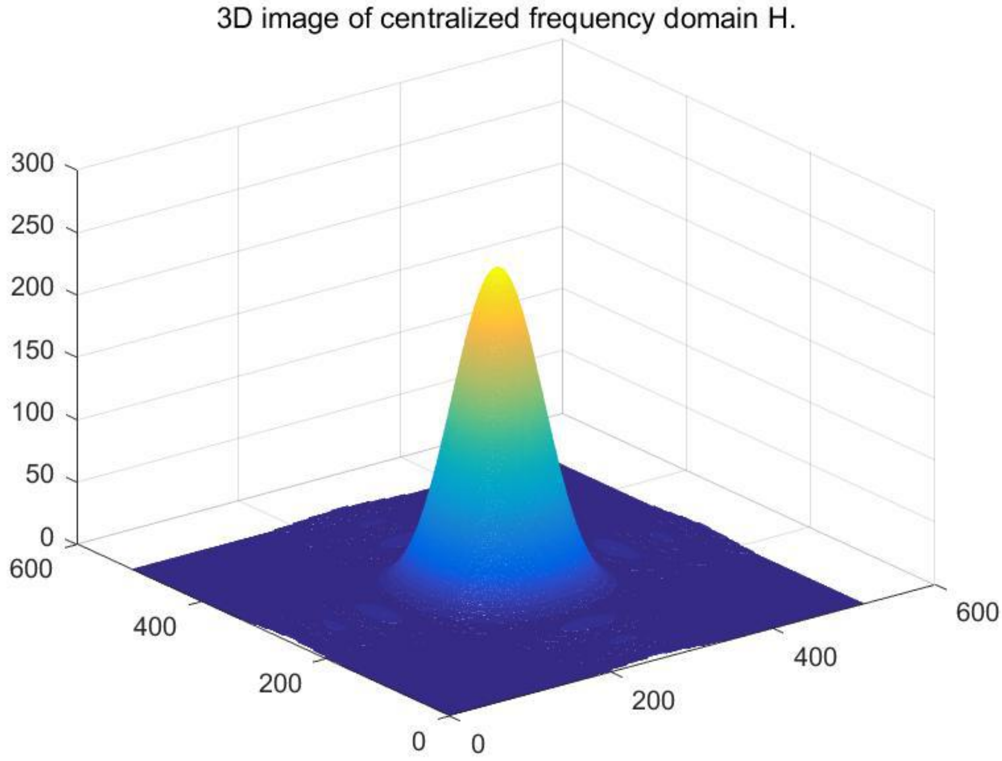


图 7 频域滤波器 3D 图像

5）、频域滤波器平面图、频域滤波结果、频谱中心化图像、

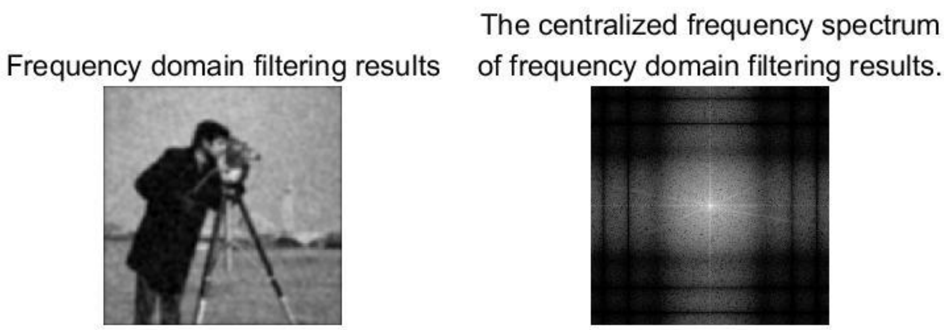


图 8 频域滤波

六、实验心得体会及建议

1、对于自己编写的傅里叶变化代码，计算机运算时间过长，故采用的 matlab 库文件自带的快速傅里叶公式。

2、频域滤波的最终图像中周围有黑边现象（相当于周围扩展了 8 行，8 列）。

3、空域和频域的实验结果的中心化频谱图不相同。

七、程序流程图及源代码

1、实际信号的频谱分析及频域滤波

源代码：

```
clc,clear;
close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
y=importdata('seismic_nsamp251_tr301_2ms.txt');
y_column=y(:,1);%The first column signal data.
[height,width]=size(y_column);
L=height;%Signal length.
T=0.002;%Sampling period.
Fs=1/(T*L);%Sampling frequency.
f=1:height;
f_1=((1:height)-ceil((height)/2)).*Fs;%The two-sided frequency domain.
f_2=(0:ceil((height-1)/2)).*Fs;%The single-sided frequency domain.
t=1800:2:2300;%The time domain.
figure,subplot(3,1,1),plot(t,y_column),title('The first column
signal');
                                       %The first column signal.
xlabel('t/(ms)');
ylabel('f(t)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
y_column_z=y_column;
for i=1:height%Spectrum centralization.
    y_column_z(i,1)=y_column(i,1)*(-1)^(1+i);
end
Yf=fft(y_column_z,L);
Y_fft=abs(Yf);%Fourier transform.
Y_fft_2=Y_fft(ceil(height/2):height);%The single-sided amplitude
spectrum.
subplot(3,1,2),plot(f_1,Y_fft),title('The two-sided ampliude
spectrum.');
xlabel('u/(Hz)');
ylabel('F(u)');
```

```
subplot(3,1,3),plot(f_2,Y_fft_2),...
              title('The single-sided ampliude spectrum.');
xlabel('u/(Hz)');
ylabel('F(u)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
N1=1;
N2=2;
N4=4;
L=height;%Signal data length.
u=f;%
D0=100;
D=u-L/2;
H1=1./(1+(D/D0).^(2*N1));%n=1
H2=1./(1+(D/D0).^(2*N2));%n=2
H4=1./(1+(D/D0).^(2*N4));%n=4
figure,subplot(3,1,1),plot(u,H1),...
   title('n=1,BLPF frequency-response curve.');
xlabel('u/(Hz)');
ylabel('|H1|');
subplot(3,1,2),plot(u,H2),title('n=2,BLPF frequency-response curve.');
xlabel('u/(Hz)');
ylabel('|H2|');
subplot(3,1,3),plot(u,H4),title('n=4,BLPF frequency-response curve.');
xlabel('u/(Hz)');
ylabel('|H4|');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
D0_5=5;
D0_20=20;
D0_50=50;
H2_5=(1./(1+(D/D0_5).^(2*N2)))';%n=2
H2_20=(1./(1+(D/D0_20).^(2*N2)))';%n=2
H2_50=(1./(1+(D/D0_50).^(2*N2)))';%n=2
G_5=Yf.*H2_5;
G_20=Yf.*H2_20;
G_50=Yf.*H2_50;
g_5=real(ifft(G_5));%Inverse fft.
g_20=real(ifft(G_20));
g_50=real(ifft(G_50));
for i=1:height
   g_5(i,1)=g_5(i,1)*(-1)^(1+i);
end
for i=1:height
```

```
    g_20(i,1)=g_20(i,1)*(-1)^(1+i);
end
for i=1:height
    g_50(i,1)=g_50(i,1)*(-1)^(1+i);
end
figure,subplot(4,1,1),plot(t,y_column),...
    title('The first column signal data.');
xlabel('t/(ms)');
ylabel('f(t)');
subplot(4,1,2),plot(t,g_5),title('D0=5,n=2');
xlabel('t/(ms)');
ylabel('g5');
subplot(4,1,3),plot(t,g_20),title('D0=20,n=2');
xlabel('t/(ms)');
ylabel('g20');
subplot(4,1,4),plot(t,g_50),title('D0=50,n=2');
xlabel('t/(ms)');
ylabel('g50');
```

2、验证空频域滤波结果的一致性

源代码：

```
clc,clear;
close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
f0_u=imread('cameraman.tif');
[height,width]=size(f0_u);
r=normrnd(0,sqrt(0.02),height,width).*255;%Produce gaussian white
noise.
f0_d=double(f0_u);
f_d=f0_d+r;%Image corrupted with gaussian white noise.
f_u=uint8(f_d);%8 bit image data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%h=fspecial('gaussian',[9,9],2); %Gaussian lowpass filter of size 9*9.
hp=zeros(9,9);%Unnormalized frequency domain filter.
for i=-4:4
    for j=-4:4
        hp(i+5,j+5)=exp((-(i^2+j^2))/(2*2^2));
    end
end
h_sum=sum(sum(hp));
h=hp./h_sum;%Normalization frequency domain filter.
```

```
%%%h=fspecial('gaussian',[9,9],2);%Gaussian lowpass filter of size 9*9.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%Repetition.
f_d_r1=[f_d(:,1:4),f_d,f_d(:,width-3:width)];
f_d_r=[f_d_r1(1:4,:);f_d_r1;f_d_r1(height-3:height,:)];
%figure(100),imshow(uint8(f_d_r));
%%%Spatial filtering.
N=9;
f_L=f_d_r;
chip=zeros(3,3);
for i=1:height%Convolution.
    for j=1:width
        for m=1:N
            for n=1:N
                chip(m,n)=f_d_r(i+m-1,j+n-1).*h(N+1-m,N+1-n);
            end
        end
        s=sum(sum(chip));
        f_L(i+(N-1)/2,j+(N-1)/2)=s;
    end
end
f_Last=f_L(5:height+4,5:width+4);
f_L_u=uint8(f_Last);%8 bit image data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%Spectrum centralization.
M=2*height;
N=2*width;
f_centralization=zeros(M,N);
f_centralization(1:height,1:width)=f_d;
for i=1:M
    for j=1:N
        f_centralization(i,j)=f_centralization(i,j).*(-1)^(i+j);
    end
end
%%%Frequency domain filter.
%%%Pad with zeros.&Spectrum centralization.
h_centralization=zeros(M,N);
h_centralization(1:9,1:9)=h;
for i=1:M
    for j=1:N
        h_centralization(i,j)=h_centralization(i,j).*(-1)^(i+j);
    end
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%Two-dimension fourier transform.of h&f
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
H=fft2(h_centralization);
F=fft2(f_centralization);
G=F.*H;
g=real(ifft2(G));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%Inverse frequency centralization.
for i=1:M
    for j=1:N
        g(i,j)=g(i,j).*(-1)^(i+j);
    end
end
g_L=g(5:height+4,5:width+4);
g_L_u=uint8(g_L);%Image filtered in frequency domain.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
figure(1),subplot(2,2,1),imshow(f0_u),title('Original image.');
subplot(2,2,2),imshow(f_u),...
    title('Image corrupted with gaussian white noise.');
subplot(2,2,3),imshow(f_L_u),title('Spatial domain filtering
results');
subplot(2,2,4),imshow(g_L_u),title('Frequency domain filtering
results');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%The centralized frequency spectrum of original image.
f0_centralization=zeros(M,N);
f0_centralization(1:height,1:width)=f0_d;
for i=1:M
    for j=1:N
        f0_centralization(i,j)=f0_centralization(i,j).*(-1)^(i+j);
    end
end
F0=fft2(f0_centralization);
```

```
F0_u=uint8(mat2gray(log(1+abs(F0)))*255);%The centralized frequency
spectrum of original image.
F_u=uint8(mat2gray(log(1+abs(F)))*255);%The centralized frequency
spectrum of image...
                ...corrupted with gaussian white noise..
h_u=uint8(mat2gray(h)*255);%The plane graph of spatial filter h.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%The centralized frequency spectrum of spatial domain filtering
results.
f_Last_centralization=zeros(M,N);
f_Last_centralization(1:height,1:width)=f_Last;
for i=1:M
    for j=1:N

f_Last_centralization(i,j)=f_Last_centralization(i,j).*(-1)^(i+j);
    end
end
F_Last=fft2(f_Last_centralization);
F_Last_u=uint8(mat2gray(log(1+abs(F_Last)))*255);%The centralized
frequency ...
               ...spectrum of spatial domain filtering results.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%The centralized frequency spectrum image of frequency domain filtering
%%%results.
G_u=uint8(mat2gray(log(1+abs(G)))*255);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
figure(2),subplot(2,2,1),imshow(F0_u),...
    title('Original image.');
subplot(2,2,2),imshow(F_u),...
title('Image corrupted with gaussian white noise.');
subplot(2,2,3),imshow(F_Last_u),title({'The centralized frequency
spectrum'
'of spatial domain filtering results.'});
subplot(2,2,4),imshow(G_u),title({'The centralized frequency spectrum'
'of frequency domain filtering results.'});
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
figure(3),subplot(2,2,[1,2]),imshow(h_u),...
    title('The plane image of spatial filter h.');
subplot(2,2,3),imshow(f_L_u),title('Spatial domain filtering
results');
```

```
subplot(2,2,4),imshow(F_Last_u),title({'The centralized frequency
spectrum'
'of spatial domain filtering results.'});
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%The plane image and 3D image of centralized frequency H.
H_u=uint8(mat2gray(abs(H))*255);
dx=1:M;
dy=1:N;
[X,Y]=meshgrid(dx,dy);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
figure(4),mesh(X,Y,H_u),title...
    ('3D image of centralized frequency domain H.');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
figure(5),subplot(2,2,[1,2]),imshow(H_u),title('The plane image of H');
subplot(2,2,3),imshow(g_L_u),title('Frequency domain filtering
results');
subplot(2,2,4),imshow(G_u),title({'The centralized frequency spectrum'
'of frequency domain filtering results.'});
```

八、思考题

1、分别阐述和解释什么叫信号的采样频率、奈奎斯特（Nyquist）频率、时间采样率及频率分辨率？

采样频率：

采样速率，每秒钟从连续信号中提取并组成离散信号的采样个数。

$$f_0 = \frac{1}{T} = \frac{1}{N * dt}$$

N 为信号长度，dt 为信号时间间隔。

奈奎斯特频率：

采样频率的最小值，即临界采样频率。

时间采样率：未听说过。

频率分辨率：

采样能分辨的最大频率。

2、简要叙述频率滤波与时域滤波在处理上有什么不同。

频域滤波：

先傅里叶变换，在相乘，滤波。没有卷积一项。

时域滤波：

直接卷积，不涉及任何变换。

3、试说明数字图像频域滤波的优势。

易于电路实现。

4、数字图像的频域滤波中，为什么原始图像和对应的滤波器均需要采取补零延拓数据。

防止频率纠缠。