《数字图像处理》实验报告

实验名称　　：实验 8 数字图像的频域转换

实验日期　　：2022.10.21

姓　　名　　：傅康

学　　号　　：084520126

班　　级　　：医信 20

成　　绩　　：

信息技术学院

南京中医药大学

实验目的：

1. 掌握 MATLAB 工具箱实现图像的傅里叶变换
2. 掌握 MATLAB 工具箱实现图像的离散余弦变换
3. 掌握频谱图的逆变换方法，观察逆变换后的图像。

实验内容和要求

建立一个名为“xxxxx 实验 8”的解决方案（xxxxx 为自己的学号）
数据准备：一个灰度图像文件，一个 8*8 的 uint8 的二维矩阵

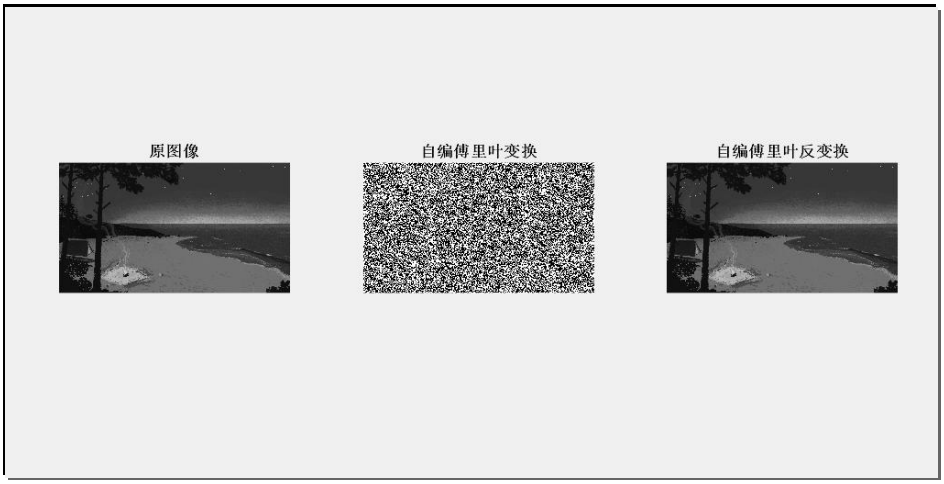
1. 自主编程实现二维图像的离散傅立叶变换及傅立叶反变换
2. 利用 fft2 ()函数及 ifft2 ()函数实现图像及矩阵的傅里叶正变换和反变换，并与自主编程实现的频域变换结果及执行时间做对比；
3. 掌握 fftshift()、ifftshift()等函数的使用。
4. 掌握傅立叶频谱得到幅度谱和相位谱的处理方法。
5. 对不同频率的正弦光栅化图像进行频域转换，对比正弦光栅化图像与幅度谱及相位谱之间的属性关系。

运行结果（写清题号）

描述实验的基本步骤，用数据和图片给出各个步骤中取得的实验结果和源代码，并进行必要的讨论，必须包括原始图像及其计算/处理后的图像以及相应的解释。

1. 自主编程实现二维图像的离散傅立叶变换及傅立叶反变换

利用二维离散傅里叶变换的可分性，可以先一维变换嵌套进行实现，即先对每一列进行一维 DFT，再对中间结果的每一行进行一维 DFT 得到二维 DFT，也可以先行后列。反变换逆过来即可。F=Gm*f*Gn, $f=Gm^{-1}*F*Gn^{-1}$。



```
subplot(1,3,1),imshow(A),title('原图像');

size_A = size(A);
length = size_A(1);  % 图(矩阵)的长
width = size_A(2);  % 图(矩阵)的宽

% 傅里叶正变换相关矩阵:
Wm = exp(-1i*2*pi/length);% 不同G中用不同的W
Wn = exp(-1i*2*pi/width);
Em = zeros(length); % E是辅助矩阵
En = zeros(width);
Gm = zeros(length)+Wm; % G是计算时用的矩阵
Gn = zeros(width)+Wn;
F = zeros(length,width);

%%%傅里叶变换%%%
% 对Gm的计算: 即先对f(x,y)的每一列进行一维DFT
for row = 0:length-1
    for col = 0:length-1
        Em(row+1,col+1) = row * col;%ux
        Gm(row+1,col+1) = Gm(row+1,col+1)^Em(row+1,col+1);
    end
end
% 对Gn的计算: 即先对f(x,y)的每一行进行一维DFT
for row = 0:width-1
    for col = 0:width-1
        En(row+1,col+1) = row * col;%yv
        Gn(row+1,col+1) = Gn(row+1,col+1)^En(row+1,col+1);
    end
end
% F = Gm*f*Gn,一般画图只要实部, 若作为输入则实虚都要!
F = Gm*A*Gn;
subplot(1,3,2),imshow(real(F)),title('自编傅里叶变换');

%%%傅里叶反变换%%%
Em = zeros(length);
En = zeros(width);      % E是辅助矩阵
Gm = zeros(length)+Wm;
Gn = zeros(width)+Wn;  % G是计算时用的矩阵
f = zeros(length,width);

for row = 0:length-1
    for col = 0:length-1
        Em(row+1,col+1) = -row * col;%指数为正,所以要加个-号,和Wm、Wn负负为正
        Gm(row+1,col+1) = Gm(row+1,col+1)^Em(row+1,col+1);
    end
end
Gm = Gm/length;
for row = 0:width-1
    for col = 0:width-1
        En(row+1,col+1) = -row * col;
        Gn(row+1,col+1) = Gn(row+1,col+1)^En(row+1,col+1);
    end
end
Gn = Gn/width;   % 注意:算完G后/width /length

f = Gm*F*Gn;
subplot(1,3,3),imshow(real(f)),title('自编傅里叶反变换');
```

2. 利用 fft2 ()函数及 ifft2 ()函数实现图像及矩阵的傅里叶正变换和反变换，并与自主编程实现的频域变换结果及执行时间做对比；

图像：代码可以基于 1.进行增加和修改，此处只填写重要的更新（自带函数和比较）代码。由输出结果可看出 matlab 自带的处理函数比自编的处理函数效率高了非常多，对于像素非常多的图片处理速度很快。

```
%fft2 ()函数
self_contained = fft2(A);   % matlab自带函数对比
subplot(2,3,2);imshow(real(self_contained));
title('fft2');
```

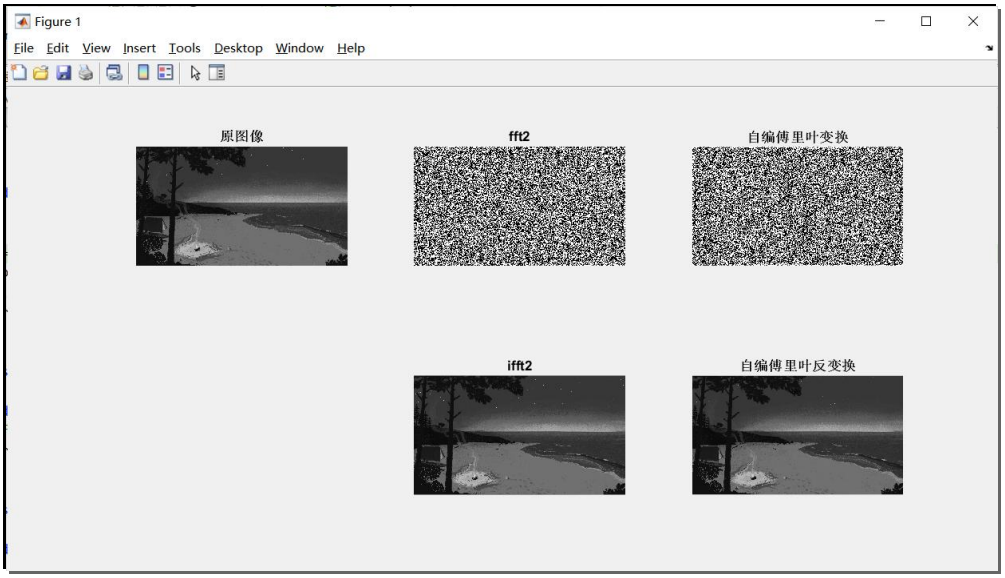
```
%%%傅里叶反变换%%%
self_containedB = ifft2(F);
subplot(2,3,5),imshow(real(self_containedB)),title('ifft2');
```

```
%结果比较
error = sum(sum((real(F)-real(self_contained)).^2));%误差
if error < 10^(-7)
    fprintf('fft2与自编结果一致!\n');
else
    fprintf('fft2与自编结果不一致!\n');
end
```

```
%反变换后与原图结果比较
error = sum(sum((real(f)-real(A)).^2));
if error < 10^(-10)
    fprintf('反变换后与原图一致!\n');
else
    fprintf('反变换后与原图不一致!\n');
end
%ifft2与自编反变换结果比较
error = sum(sum((real(f)-real(self_containedB)).^2));
if error < 10^(-10)
    fprintf('ifft2与自编结果一致\n');
else
    fprintf('ifft2与自编结果不一致!\n');
end
```

```
disp(['fft2耗时: ',num2str(time_fft2)]);
disp(['myfft2耗时: ',num2str(time_myfft2)]);
disp(['ifft2耗时: ',num2str(time_ifft2)]);
disp(['myifft2耗时: ',num2str(time_myifft2)]);
```

```
>> ex8_1_2
fft2与自编结果一致!
反变换后与原图一致!
ifft2与自编结果一致
fft2耗时: 0.17115
myfft2耗时: 18.3703
ifft2耗时: 0.17566
myifft2耗时: 19.1654
```



矩阵：

原矩阵

8x8 double

	1	2	3	4	5	6	7	8
1	0.8941	0.8157	0.3529	0.3804	0.5686	0.1647	0.2275	0.1059
2	0.9608	0.2431	0.8314	0.5686	0.4706	0.6039	0.9137	0.9647
3	0.5490	0.9294	0.5843	0.0745	0.0118	0.2627	0.1529	0.0039
4	0.1373	0.3490	0.5490	0.0510	0.3373	0.6549	0.8275	0.7765
5	0.1490	0.1961	0.9176	0.5294	0.1608	0.6902	0.5373	0.8196
6	0.2549	0.2510	0.2863	0.7804	0.7961	0.7490	1	0.8706
7	0.8431	0.6157	0.7569	0.9373	0.3098	0.4510	0.0784	0.0824
8	0.2549	0.4745	0.7529	0.1294	0.5294	0.0824	0.4431	0.4000

fft2

8x8 complex double

	1	2	3	4	5	6	7	8
1	31.4471 + 0.0000i	1.4162 - 0.5986i	-1.9843 - 0.0588i	0.3015 + 1.1033i	1.4314 + 0.0000i	0.3015 - 1.1033i	-1.9843 + 0.0588i	1.4162 + 0.5986i
2	-0.5235 + 0.6684i	2.8444 - 4.1041i	1.1715 + 0.0697i	0.9042 - 0.8700i	2.4502 - 0.2919i	-0.3836 + 1.3173i	3.4657 + 1.7227i	0.6904 + 0.5132i
3	0.8667 - 3.7961i	0.5669 + 2.3516i	1.1373 + 0.3647i	-1.4651 - 0.9138i	0.1765 + 0.4314i	-0.4258 - 1.9908i	-1.9451 - 1.8941i	-1.7036 - 1.1411i
4	-0.4569 - 2.3433i	-0.0885 + 0.4319i	2.7225 - 1.1792i	0.3321 - 3.4217i	-0.3561 - 0.0409i	-1.1453 + 0.2466i	0.7500 + 0.9952i	-0.4557 - 0.3690i
5	-3.1412 + 0.0000i	1.4634 - 5.3170i	1.7412 - 2.3255i	2.3563 - 0.5562i	-1.3608 + 0.0000i	2.3563 + 0.5562i	1.7412 + 2.3255i	1.4634 + 5.3170i
6	-0.4569 + 2.3433i	-0.4557 + 0.3690i	0.7500 - 0.9952i	-1.1453 - 0.2466i	-0.3561 + 0.0409i	0.3321 + 3.4217i	2.7225 + 1.1792i	-0.0885 - 0.4319i
7	0.8667 + 3.7961i	-1.7036 + 1.1411i	-1.9451 + 1.8941i	-0.4258 + 1.9908i	0.1765 - 0.4314i	-1.4651 + 0.9138i	1.1373 - 0.3647i	0.5669 - 2.3516i
8	-0.5235 - 0.6684i	0.6904 - 0.5132i	3.4657 - 1.7227i	-0.3836 - 1.3173i	2.4502 + 0.2919i	0.9042 + 0.8700i	1.1715 - 0.0697i	2.8444 + 4.1041i

自编傅里叶变换

8x8 complex double

	1	2	3	4	5	6	7	8
1	31.4471 + 0.0000i	1.4162 - 0.5986i	-1.9843 - 0.0588i	0.3015 + 1.1033i	1.4314 - 0.0000i	0.3015 - 1.1033i	-1.9843 + 0.0588i	1.4162 + 0.5986i
2	-0.5235 + 0.6684i	2.8444 - 4.1041i	1.1715 + 0.0697i	0.9042 - 0.8700i	2.4502 - 0.2919i	-0.3836 + 1.3173i	3.4657 + 1.7227i	0.6904 + 0.5132i
3	0.8667 - 3.7961i	0.5669 + 2.3516i	1.1373 + 0.3647i	-1.4651 - 0.9138i	0.1765 + 0.4314i	-0.4258 - 1.9908i	-1.9451 - 1.8941i	-1.7036 - 1.1411i
4	-0.4569 - 2.3433i	-0.0885 + 0.4319i	2.7225 - 1.1792i	0.3321 - 3.4217i	-0.3561 - 0.0409i	-1.1453 + 0.2466i	0.7500 + 0.9952i	-0.4557 - 0.3690i
5	-3.1412 - 0.0000i	1.4634 - 5.3170i	1.7412 - 2.3255i	2.3563 - 0.5562i	-1.3608 - 0.0000i	2.3563 + 0.5562i	1.7412 + 2.3255i	1.4634 + 5.3170i
6	-0.4569 + 2.3433i	-0.4557 + 0.3690i	0.7500 - 0.9952i	-1.1453 - 0.2466i	-0.3561 + 0.0409i	0.3321 + 3.4217i	2.7225 + 1.1792i	-0.0885 - 0.4319i
7	0.8667 + 3.7961i	-1.7036 + 1.1411i	-1.9451 + 1.8941i	-0.4258 + 1.9908i	0.1765 - 0.4314i	-1.4651 + 0.9138i	1.1373 - 0.3647i	0.5669 - 2.3516i
8	-0.5235 - 0.6684i	0.6904 - 0.5132i	3.4657 - 1.7227i	-0.3836 - 1.3173i	2.4502 + 0.2919i	0.9042 + 0.8700i	1.1715 - 0.0697i	2.8444 + 4.1041i

ifft2

8x8 complex double

	1	2	3	4	5	6	7	8
1	0.8941 - 0.0000i	0.8157 - 0.0000i	0.3529 - 0.0000i	0.3804 - 0.0000i	0.5686 - 0.0000i	0.1647 - 0.0000i	0.2275 - 0.0000i	0.1059 - 0.0000i
2	0.9608 - 0.0000i	0.2431 - 0.0000i	0.8314 - 0.0000i	0.5686 - 0.0000i	0.4706 - 0.0000i	0.6039 - 0.0000i	0.9137 + 0.0000i	0.9647 + 0.0000i
3	0.5490 - 0.0000i	0.9294 + 0.0000i	0.5843 - 0.0000i	0.0745 - 0.0000i	0.0118 - 0.0000i	0.2627 - 0.0000i	0.1529 - 0.0000i	0.0039 - 0.0000i
4	0.1373 - 0.0000i	0.3490 - 0.0000i	0.5490 - 0.0000i	0.0510 - 0.0000i	0.3373 - 0.0000i	0.6549 + 0.0000i	0.8275 + 0.0000i	0.7765 + 0.0000i
5	0.1490 - 0.0000i	0.1961 - 0.0000i	0.9176 + 0.0000i	0.5294 - 0.0000i	0.1608 - 0.0000i	0.6902 + 0.0000i	0.5373 + 0.0000i	0.8196 + 0.0000i
6	0.2549 - 0.0000i	0.2510 - 0.0000i	0.2863 - 0.0000i	0.7804 + 0.0000i	0.7961 + 0.0000i	0.7490 + 0.0000i	1.0000 + 0.0000i	0.8706 + 0.0000i
7	0.8431 + 0.0000i	0.6157 + 0.0000i	0.7569 + 0.0000i	0.9373 + 0.0000i	0.3098 + 0.0000i	0.4510 + 0.0000i	0.0784 - 0.0000i	0.0824 - 0.0000i
8	0.2549 - 0.0000i	0.4745 + 0.0000i	0.7529 + 0.0000i	0.1294 - 0.0000i	0.5294 + 0.0000i	0.0824 - 0.0000i	0.4431 + 0.0000i	0.4000 + 0.0000i

自编傅里叶反变换

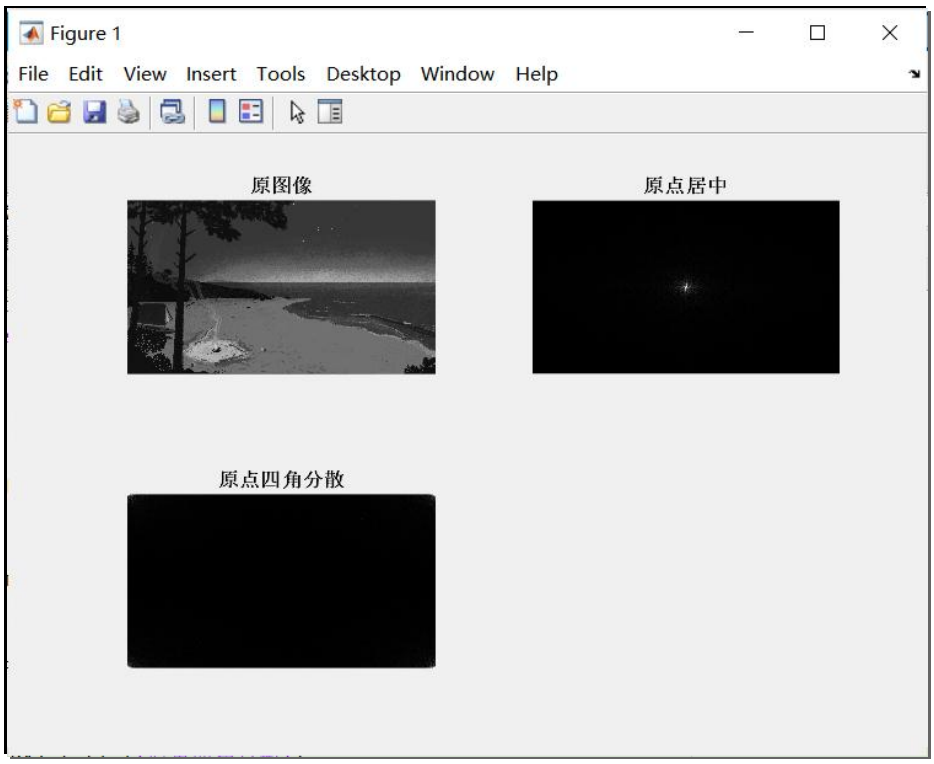
8x8 complex double

	1	2	3	4	5	6	7	8
1	0.8941 - 0.0000i	0.8157 - 0.0000i	0.3529 - 0.0000i	0.3804 - 0.0000i	0.5686 - 0.0000i	0.1647 - 0.0000i	0.2275 - 0.0000i	0.1059 + 0.0000i
2	0.9608 - 0.0000i	0.2431 - 0.0000i	0.8314 - 0.0000i	0.5686 - 0.0000i	0.4706 - 0.0000i	0.6039 - 0.0000i	0.9137 + 0.0000i	0.9647 + 0.0000i
3	0.5490 - 0.0000i	0.9294 - 0.0000i	0.5843 - 0.0000i	0.0745 - 0.0000i	0.0118 + 0.0000i	0.2627 + 0.0000i	0.1529 + 0.0000i	0.0039 + 0.0000i
4	0.1373 - 0.0000i	0.3490 - 0.0000i	0.5490 - 0.0000i	0.0510 - 0.0000i	0.3373 - 0.0000i	0.6549 + 0.0000i	0.8275 + 0.0000i	0.7765 + 0.0000i
5	0.1490 - 0.0000i	0.1961 - 0.0000i	0.9176 - 0.0000i	0.5294 - 0.0000i	0.1608 + 0.0000i	0.6902 + 0.0000i	0.5373 + 0.0000i	0.8196 + 0.0000i
6	0.2549 - 0.0000i	0.2510 - 0.0000i	0.2863 - 0.0000i	0.7804 - 0.0000i	0.7961 + 0.0000i	0.7490 + 0.0000i	1.0000 + 0.0000i	0.8706 + 0.0000i
7	0.8431 + 0.0000i	0.6157 + 0.0000i	0.7569 + 0.0000i	0.9373 + 0.0000i	0.3098 + 0.0000i	0.4510 + 0.0000i	0.0784 + 0.0000i	0.0824 + 0.0000i
8	0.2549 + 0.0000i	0.4745 + 0.0000i	0.7529 + 0.0000i	0.1294 + 0.0000i	0.5294 + 0.0000i	0.0824 + 0.0000i	0.4431 + 0.0000i	0.4000 + 0.0000i

由矩阵可看出，自编和系统自带的傅里叶(反)变换函数，结果都一样，反变换后矩阵又回到原矩阵。

3. 掌握 fftshift()、ifftshift()等函数的使用。

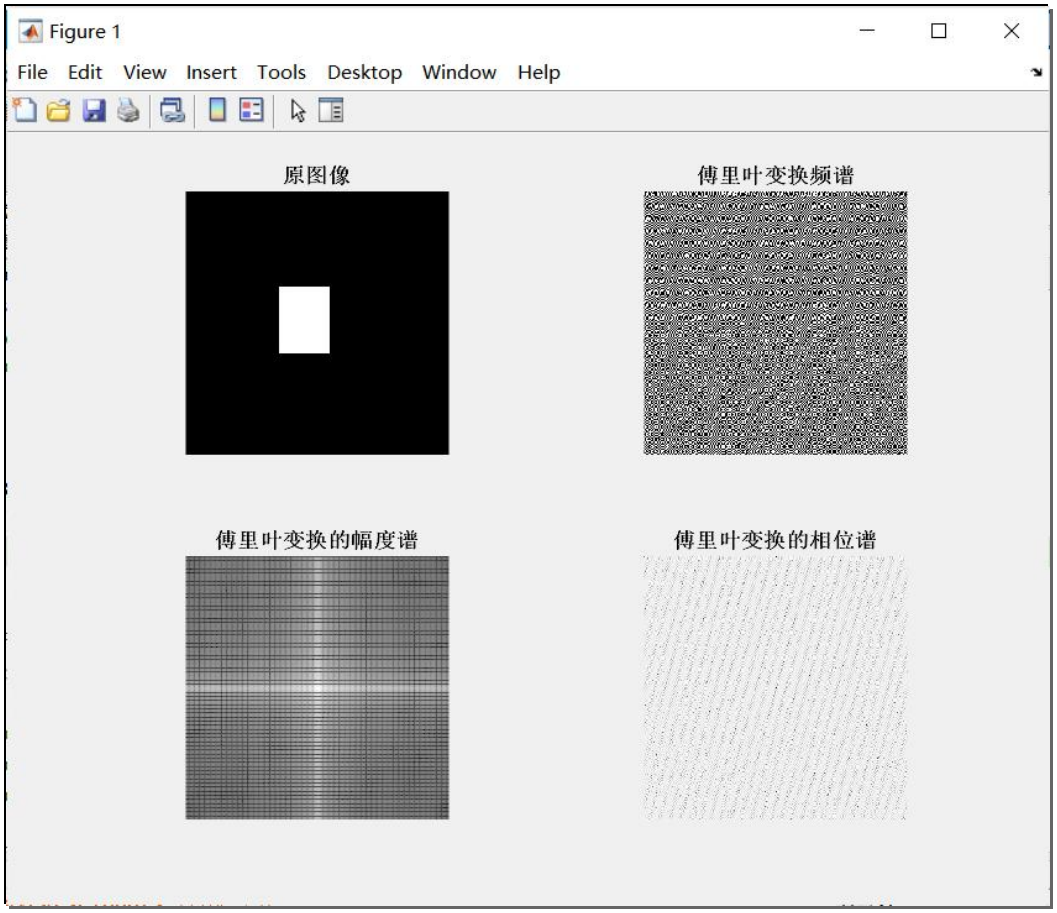
fftshift()将将零频分量移动到数组中心，原点居中，重新排列傅里叶变换 X；ifftshift()进行过零频平移的傅里叶变换 Y 重新排列回原始变换输出的样子，即撤消 fftshift 的结果,原点四角分散。



```
% 3.掌握fftshift()、ifftshift()等函数的使用。
img = imread('C:\Users\Knight6\Pictures\matlabimage\ex8\iTab-20220722.png');
img = im2double(img);
img = rgb2gray(img);% rgb转为灰度图像
subplot(221),imshow(img),title('原图像');
%fftshift（）原点移动到频率矩形的中心
fftA=fft2(img);%傅里叶变换
sfftA=fftshift(fftA);
A=abs(sfftA);
B=(A-min(min(A)))/(max(max(A))-min(min(A)))*255;%归一化
subplot(222),imshow(B),title('原点居中');
%ifftshift()
lsfftA=ifftshift(sfftA);
A1=abs(lsfftA);
B1=(A1-min(min(A1)))/(max(max(A1))-min(min(A1)))*255;
subplot(223),imshow(B1),title('原点四角分散');
```

4. 掌握傅立叶频谱得到幅度谱和相位谱的处理方法。

原图



平移

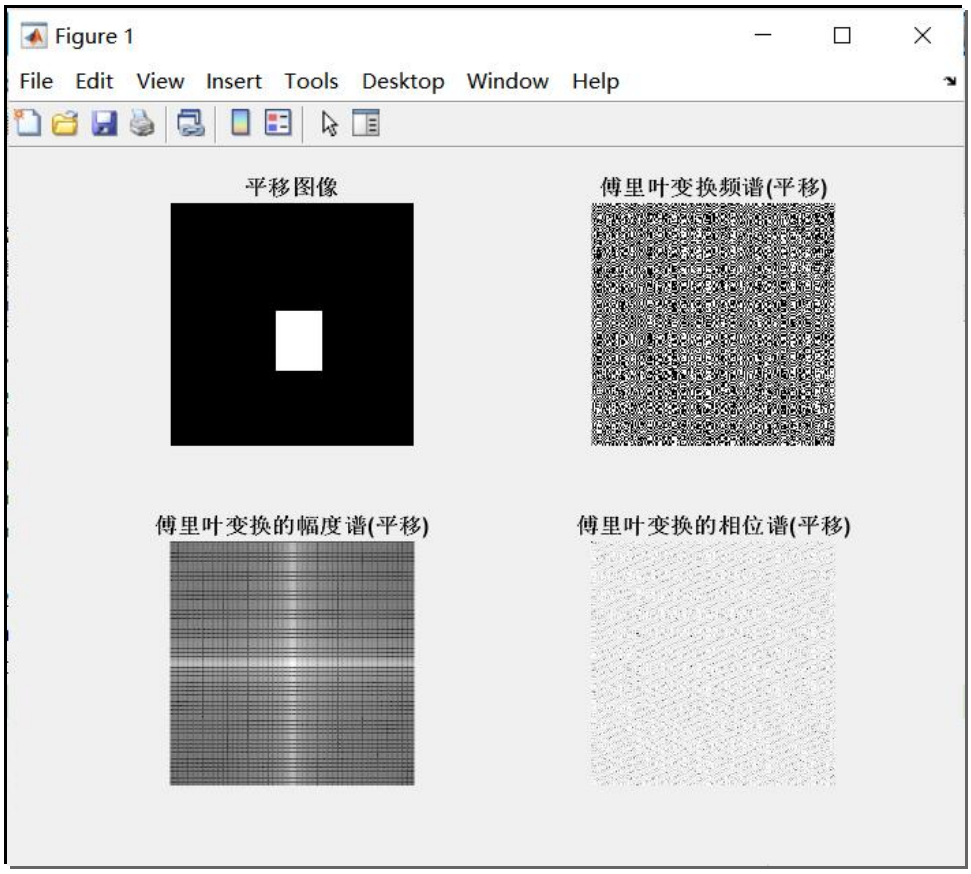


Figure 1
File Edit View Insert Tools Desktop Window Help
平移图像
傅里叶变换频谱(平移)
傅里叶变换的幅度谱(平移)
傅里叶变换的相位谱(平移)

旋转

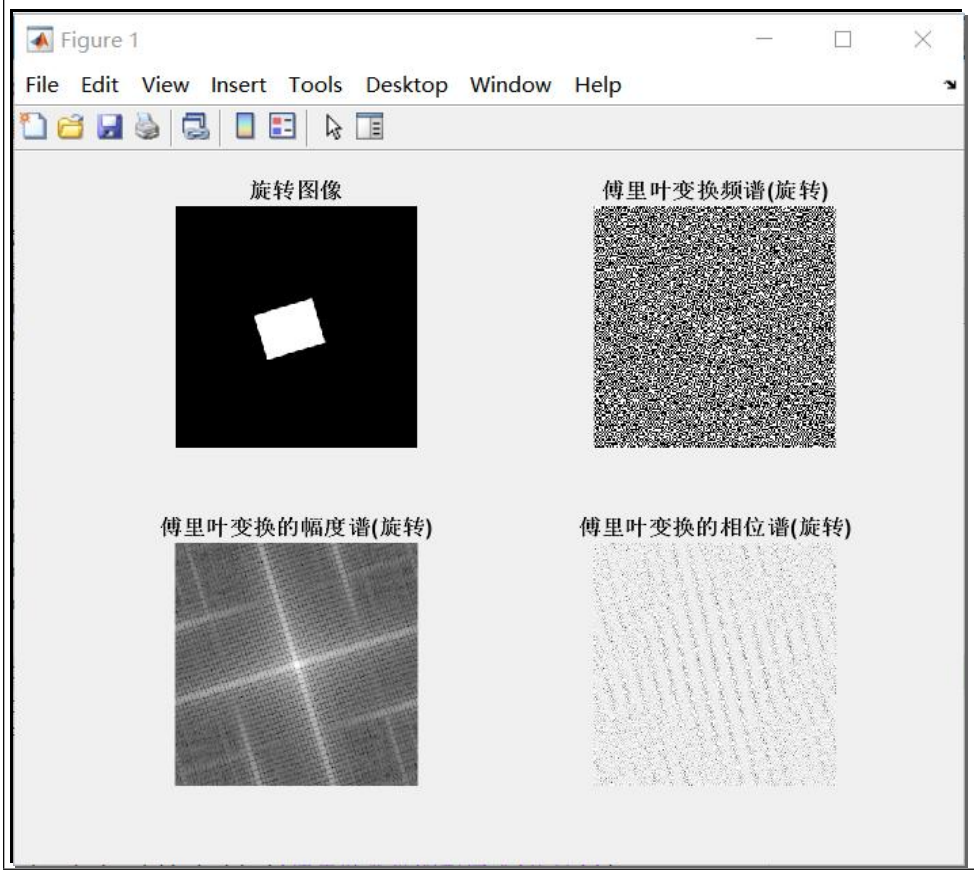
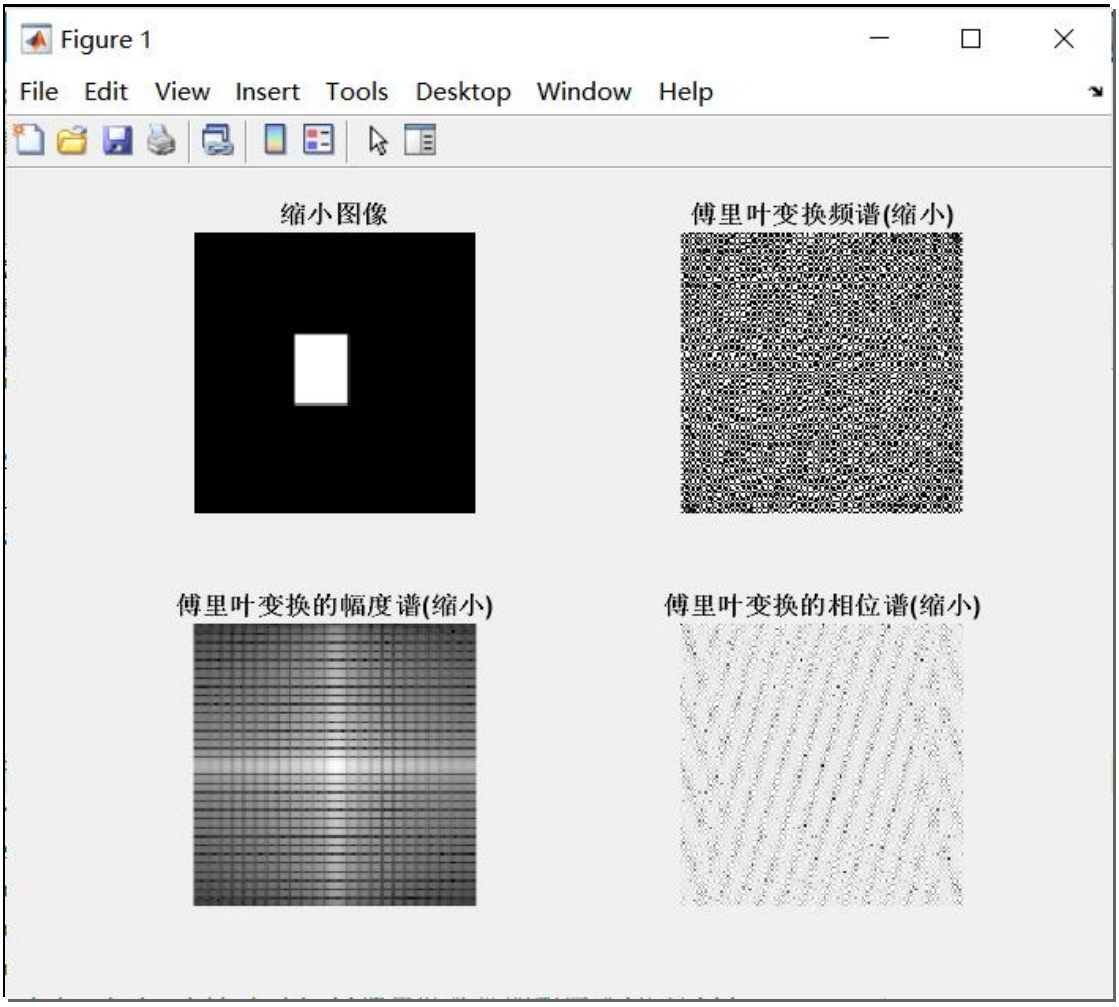


Figure 1
File Edit View Insert Tools Desktop Window Help
旋转图像
傅里叶变换频谱(旋转)
傅里叶变换的幅度谱(旋转)
傅里叶变换的相位谱(旋转)

缩小



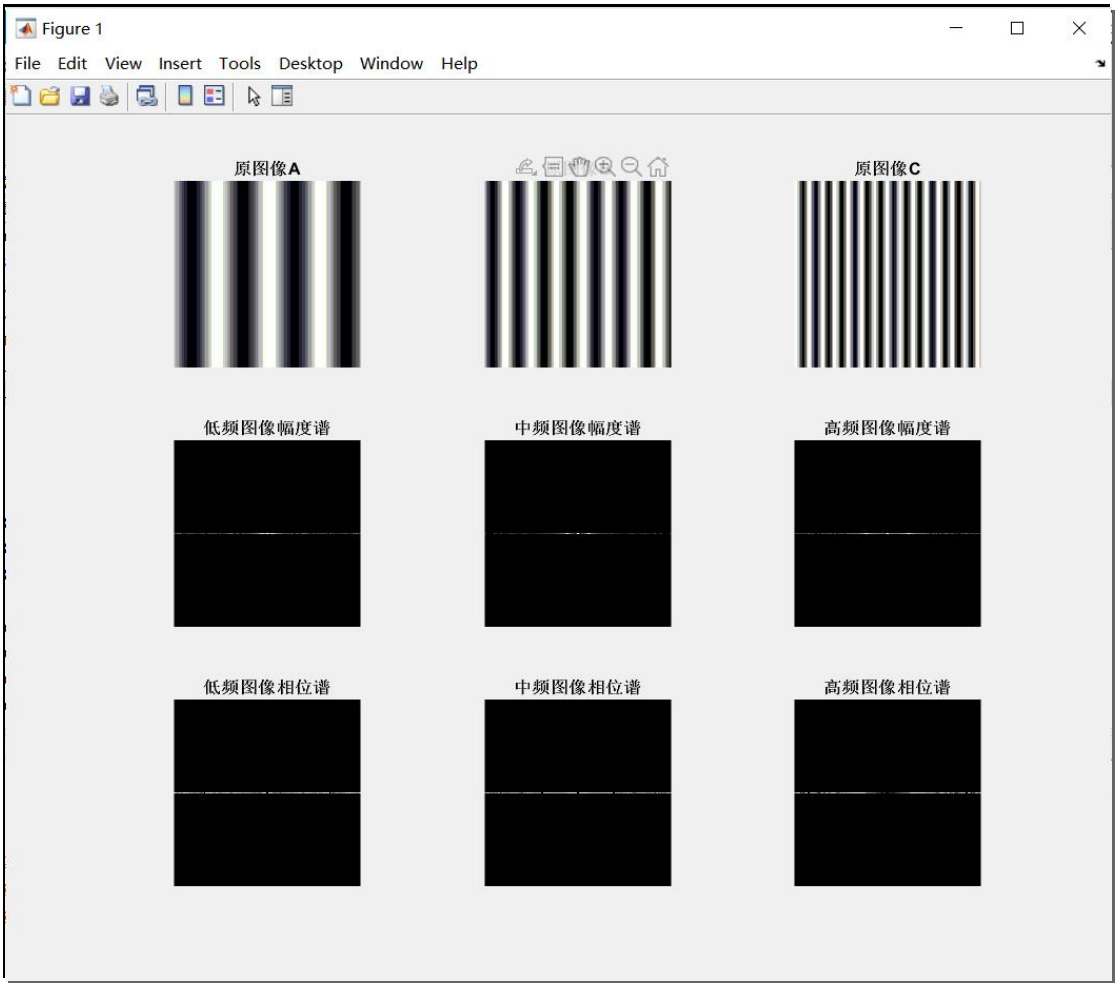
傅里叶频谱可以通过 `log(abs(fA));`和 `log(angle(fA)*180/pi);`得到幅度谱和相位谱。同时，由以上运行结果，并根据二维离散傅里叶变换的几何变换特性可以知道，图像的平移不影响其傅里叶变换的幅值，只改变相位谱。图像空域旋转，则变换域函数 DFT 也旋转同样的角度；图像缩放则导致其傅里叶变换在频域尺度的相反变换。

代码

```
% 4.掌握傅立叶频谱得到幅度谱和相位谱的处理方法。
A=imread('C:\Users\Knight6\Pictures\matlabimage\ex8\DCTtry1.jpg');%原图
A=rgb2gray(A);
subplot(221);imshow(A);title('原图像');

scale=imresize(A,0.5,'bilinear');%缩小变换
rotate=imread('C:\Users\Knight6\Pictures\matlabimage\ex8\DCTtry2.jpg');%旋转变换
tform=maketform('AFFINE',[1 0 0;0 1 0;20 20 1]);
trans=imtransform(A,tform,'XData',[1,size(A,2)],'YData',[1,size(A,1)]);%平移变换

% fA=fftshift(fft2(A(:,:,1)));%原图DFT
% sA=log(abs(fA));
% phA=log(angle(fA)*180/pi);
% subplot(222);imshow(fA);title('傅里叶变换频谱');
% subplot(223);imshow(sA,[]);title('傅里叶变换的幅度谱');
% subplot(224);imshow(phA,[]);title('傅里叶变换的相位谱');

% fscale=fftshift(fft2(scale(:,:,1)));%缩小DFT
% sscale=log(abs(fscale));
% phscale=log(angle(fscale)*180/pi);
% subplot(221);imshow(scale);title('缩小图像');
% subplot(222);imshow(fscale);title('傅里叶变换频谱(缩小)');
% subplot(223);imshow(sscale,[]);title('傅里叶变换的幅度谱(缩小)');
% subplot(224);imshow(phscale,[]);title('傅里叶变换的相位谱(缩小)');

% frotate=fftshift(fft2(rotate(:,:,1)));%旋转DFT
% srotate=log(abs(frotate));
% phrotate=log(angle(frotate)*180/pi);
% subplot(221);imshow(rotate);title('旋转图像');
% subplot(222);imshow(frotate);title('傅里叶变换频谱(旋转)');
% subplot(223);imshow(srotate,[]);title('傅里叶变换的幅度谱(旋转)');
% subplot(224);imshow(phrotate,[]);title('傅里叶变换的相位谱(旋转)');
%
ftrans=fftshift(fft2(trans(:,:,1)));%平移DFT
strans=log(abs(ftrans));
phtrans=log(angle(ftrans)*180/pi);
subplot(221);imshow(trans);title('平移图像');
subplot(222);imshow(ftrans);title('傅里叶变换频谱(平移)');
subplot(223);imshow(strans,[]);title('傅里叶变换的幅度谱(平移)');
subplot(224);imshow(phtrans,[]);title('傅里叶变换的相位谱(平移)');
```

5. 对不同频率的正弦光栅化图像进行频域转换，对比正弦光栅化图像与幅度谱及相位谱之间的属性关系。

正弦波的频率越低，对称的频点越靠近中心，随着频率不断增加，对称的频点离中心越来越远。




```
imA=imread('C:\Users\Knight6\Pictures\matlabimage\ex8\DP1.jpg');%低频图像
imB=imread('C:\Users\Knight6\Pictures\matlabimage\ex8\DP2.jpg');%中频图像
imC=imread('C:\Users\Knight6\Pictures\matlabimage\ex8\DP3.jpg');%高频图像
fA =fftshift(fft2(imA(:,:,1)));
fB = fftshift(fft2(imB(:,:,1)));
fC = fftshift(fft2(imC(:,:,1)));

sA=log(abs(fA));
sB=log(abs(fB));
sC=log(abs(fC));
phA=log(angle(fA)*180/pi);
phB=log(angle(fB)*180/pi);
phC=log(angle(fC)*180/pi);
figure;
subplot(3,3,1);imshow(imA);title('原图像A');
subplot(3,3,2);imshow(imB);title('原图像B');
subplot(3,3,3);imshow(imC);title('原图像C');
subplot(3,3,4);imshow(sA,[]); title('低频图像幅度谱');
subplot(3,3,7);imshow(phA,[]);title('低频图像相位谱');
subplot(3,3,5);imshow(sB,[]);title('中频图像幅度谱');
subplot(3,3,8);imshow(phB,[]);title('中频图像相位谱');
subplot(3,3,6);imshow(sC,[]);title('高频图像幅度谱');
subplot(3,3,9);imshow(phC,[]);title('高频图像相位谱');
```

实验的体会与思考题

1. 通过实验深化理解频域图像与时域图像的对应变化。

傅立叶变换将图像在时间域和频率域之间相互转换，对看似复杂的数据换一种观察的维度，从而找出一些直观信息进行分析。信号在频域往往比在时域更加简单和直观的特性，图像中一些特定频率的噪声也可以在频域中进行去除。

2. 谈谈你所理解的频域。

频域：横轴是频率，纵轴是该频率信号的幅度图（频谱图），其描述了信号的频率结构及频率与该频率信号幅度的关系。