

数字图像处理实验报告三

药3 李宇晖 2013012470

实验目的

了解二维离散傅里叶变换的基本概念和意义

实验过程与结果

用 matlab 实现二维离散傅里叶变换（不需要实现 FFT）。用 matlab 编程合成多个不同角度、不同频率、不同相位的二维正弦波图像，以及多个不同角度、不同尺寸、不同中心位置的矩形图像。使用你实现的 2D DFT 函数计算上述图像的 2D DFT，显示幅度谱和相位谱（令原点居中）。所有图像的尺寸均设置为 256×256 像素。

首先写出 DFT 函数。输入大小为 $M \times N$ 的图像，根据下述二维离散傅里叶变换函数，可以将 DFT 变为矩阵形式，从而在 matlab 中写出 My_DFT 函数。

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

代码如下：

```
function [ DFT_output ] = My_DFT( N )
    [x, y] = meshgrid(0:(N-1), 0:(N-1));
    omega = exp(-2.0 * 1j * pi * (x.*y) / N);
end
```

之后利用 meshgrid()函数生成 256×256 的网格采样点，继而利用二维正弦波方程和 meshgrid()函数即可生成二维正弦波和矩形的图像。使用 matlab 自带的 fftshift()函数即可将图像对称。图像的频度谱与图像的幅度谱分别由下式左项与右项得到。

$$F(u, v) = |F(u, v)| e^{j\theta(u, v)}$$

频度图即为实数部分的绝对值，由 abs()函数得到。幅度图即为复数部分的 arctan 辐射角，可由 angle()函数得到。

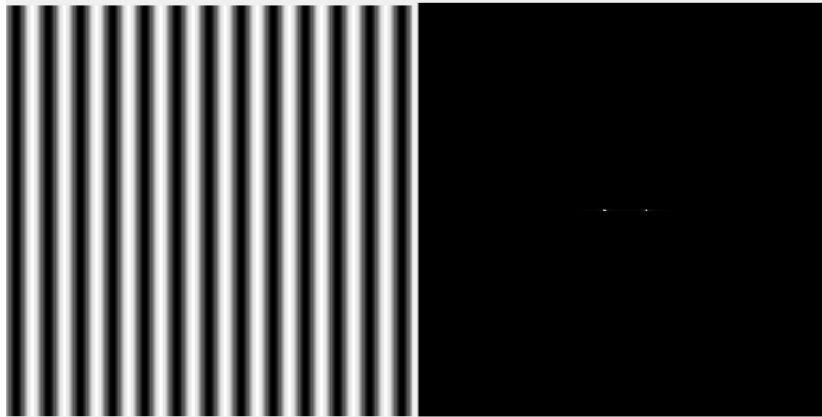
首先比较 My_DFT 函数和 matlab 工具箱中的 fftn 函数。

代码如下：

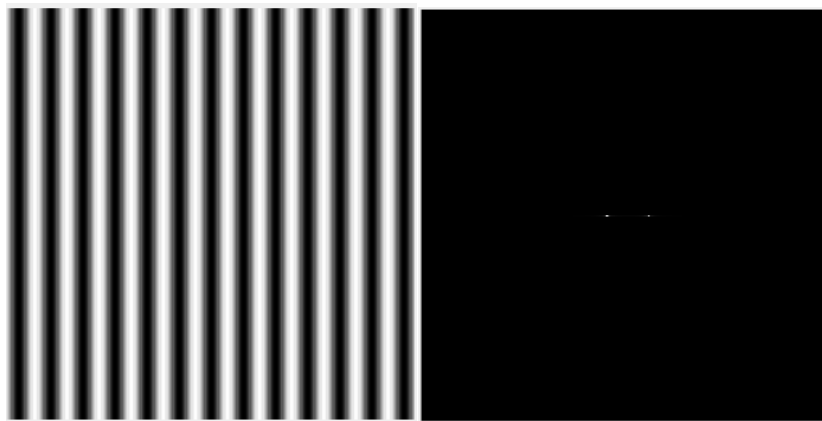
```
close all, clear all
%create some rectangle
% Construct 2D sine wave
[X, Y] = meshgrid(1:256);
f1 = cos(2*pi*0.05*X);
figure(1),
%compare the frequency of my_DFT and fft2
subplot(2,2,1),imshow(f1,[])
f1_my_DFT = My_DFT(256) * f1 * My_DFT(256);%compute and visualize the DFT of f using my_DFT
f1_my_DFT = fftshift(f1_my_DFT);%balance the image
subplot(2,2,2),imshow(abs(f1_my_DFT),[]);%show the margin
f1_DFT = fftn(f1);%use the fftn function of matlab
f1_DFT = fftshift(f1_DFT);%balance the image
```

```
subplot(2,2,3),imshow(f1,[]);
subplot(2,2,4),imshow(abs(f1_DFT),[]);%show the margin
```

结果如下：



使用 My_DFT 实现的傅里叶变换频谱



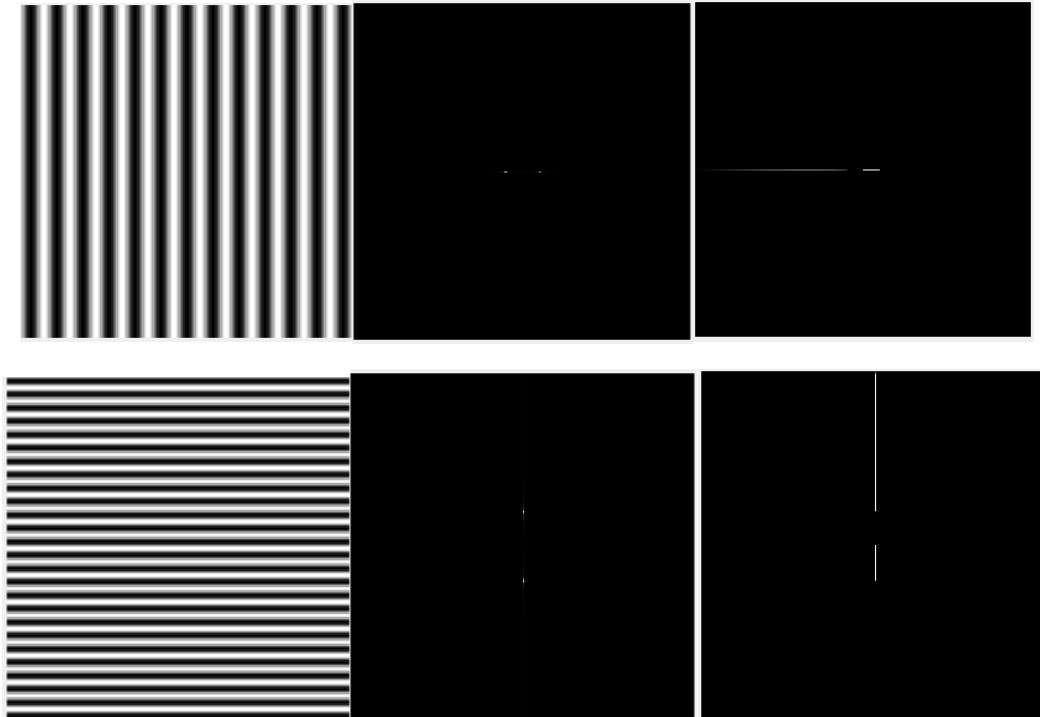
使用 fftn 函数实现的傅里叶变换频谱

可以看出，两个函数同样可以表示出分离出的正弦信号的频率。为了使变换更规范，以下均使用 `fftn` 函数进行实验。

不同频率的正弦傅里叶变换代码如下：

```
figure(2),
%construct 2D sine wave
f2 = cos(2*pi*0.1*Y);
subplot(2,3,1),imshow(f1,[]);
subplot(2,3,4),imshow(f2,[]);
% Compute and visualize the 30-by-30 DFT of f
f2_DFT = fftn(f2);%compute and visualize the DFT of f using fftn
f2_DFT = fftshift(f2_DFT); %balance
subplot(2,3,2), imshow(abs(f1_DFT),[]); %show the margin
subplot(2,3,3), imshow(angle(f1_DFT*180/pi)); %show the frequency
subplot(2,3,5), imshow(abs(f2_DFT), []);
subplot(2,3,6), imshow(angle(f2_DFT)*180/pi);
```

结果如下：

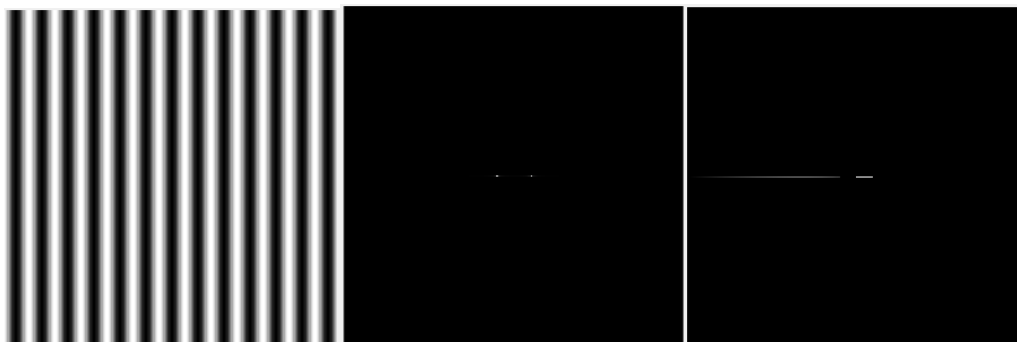


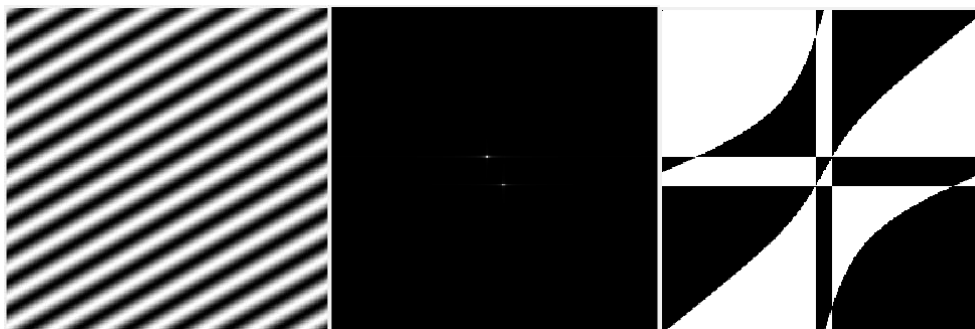
上图中，左图是空域中的原图像，中图是频谱图，右图是相位图。可以看出频谱图可以表征出图像在空域中的变化方向，相位图也能在一定程度上表现。

不同角度的正弦函数傅立叶变换代码如下：

```
figure(3),
% Compute and visualize the 30-by-30 DFT of f
f3 = cos(2*pi*0.05*(cos(pi/3)*X+sin(pi/3)*Y));
subplot(2,3,1),imshow(f1,[]);
subplot(2,3,4),imshow(f3,[]);
f3_DFT = fftn(f3);%compute and visualize the DFT of f using fftn
f3_DFT = fftshift(f3_DFT);%balance
subplot(2,3,2), imshow(abs(f1_DFT),[]); %show the margin
subplot(2,3,3), imshow(angle(f1_DFT*180/pi)); %show the frequency
subplot(2,3,5), imshow(abs(f3_DFT), []);
subplot(2,3,6), imshow(angle(f3_DFT)*180/pi);
```

结果如下：



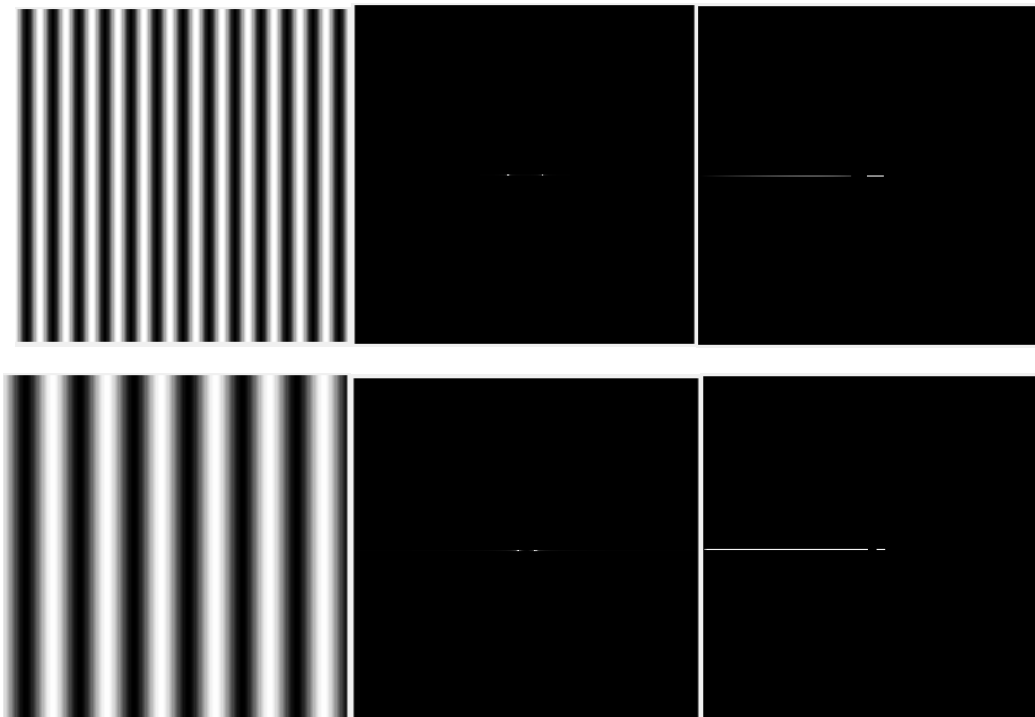


上图中，左图是空域中的原图像，中图是频谱图，右图是相位图。可以看出频谱图可以表征出图像在空域中的变化方向，但相位图相对来说难以理解。

不同相位的正弦函数傅里叶变换代码如下：

```
figure(4);
% Compute and visualize the 30-by-30 DFT of f
f3 = cos(2*pi*0.05*(cos(pi/3)*X+0.5));
subplot(2,3,1),imshow(f1,[]);
subplot(2,3,4),imshow(f3,[]);
f3_DFT = fftn(f3);%compute and visualize the DFT of f using fftn
f3_DFT = fftshift(f3_DFT);%balance
subplot(2,3,2), imshow(abs(f1_DFT),[]); %show the margin
subplot(2,3,3), imshow(angle(f1_DFT*180/pi)); %show the frequency
subplot(2,3,5), imshow(abs(f3_DFT), []);
subplot(2,3,6), imshow(angle(f3_DFT)*180/pi);
```

结果如下：



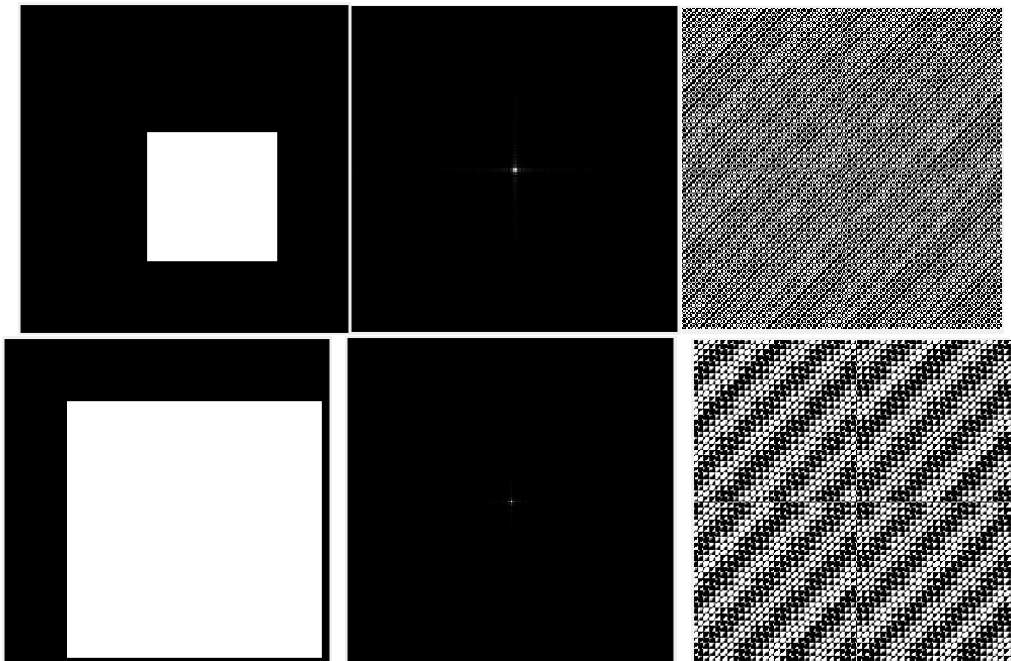
上图中，左图是空域中的原图像，中图是频谱图，右图是相位图。可以看出频谱图可以表征出图像在空域中的变化的快慢，相位图也在一定程度上表示。

不同大小的矩形图像傅里叶变换代码如下：

```
close all, clear all
initial_image = zeros(256,256);
%create some binary rectangles
[x,y] = meshgrid(100:200);
f1 = initial_image; %zero image
f1(x,y) = 1; %construct rectangle
f1_DFT = fftn(f1);%dft using fftn
f1_DFT = fftshift(f1_DFT);%balance
figure(1);
subplot(2,3,1),imshow(f1,[]);
subplot(2,3,2), imshow(abs(f1_DFT),[]); %show the margin
subplot(2,3,3), imshow(angle(f1_DFT*180/pi)); %show the frequency

[x,y] = meshgrid(50:250);
f2 = initial_image;
f2(x,y) = 1;%constrct rectangle of different size
f2_DFT = fftn(f2);%dft using fftn
f2_DFT = fftshift(f2_DFT);%balance
subplot(2,3,4),imshow(f2,[]);
subplot(2,3,5), imshow(abs(f2_DFT),[]); %show the margin
subplot(2,3,6), imshow(angle(f2_DFT*180/pi)); %show the frequency
```

结果如下：



上图中，左图是空域中的原图像，中图是频谱图，右图是相位图。可以看出频谱图可以表征出图像在空域中的变化情况，矩形越大，频域中的亮点越小。但相位图相对来说难以理解。

不同位置的矩形图像傅里叶变换代码如下：

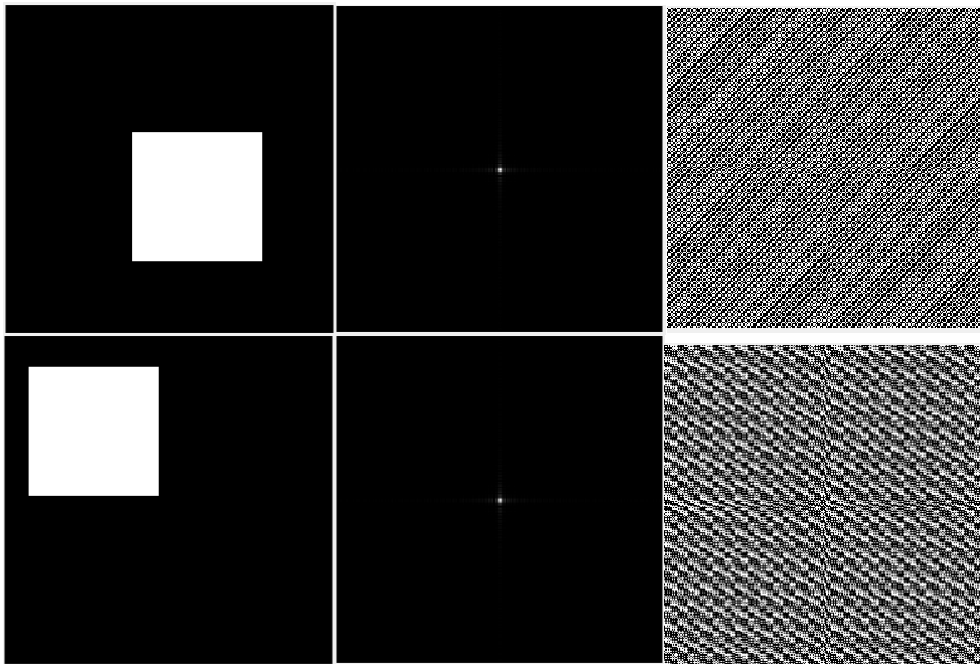
```
x=meshgrid(25:125);
```

```

y=meshgrid(20:120);
f2 = initial_image;
f2(x,y) = 1;
figure(2);
subplot(2,3,1),imshow(f1,[]);
subplot(2,3,2), imshow(abs(f1_DFT),[]); %show the margin
subplot(2,3,3), imshow(angle(f1_DFT*180/pi)); %show the frequency
f2_DFT = fftn(f2);
f2_DFT = fftshift(f2_DFT);
subplot(2,3,4),imshow(f2,[]);
subplot(2,3,5), imshow(abs(f2_DFT),[]); %show the margin
subplot(2,3,6), imshow(angle(f2_DFT*180/pi)); %show the frequency

```

结果如下：



上图中，左图是空域中的原图像，中图是频谱图，右图是相位图。可以看出在居中之后，两者的频谱图几乎一致。但相位图相对来说难以理解。

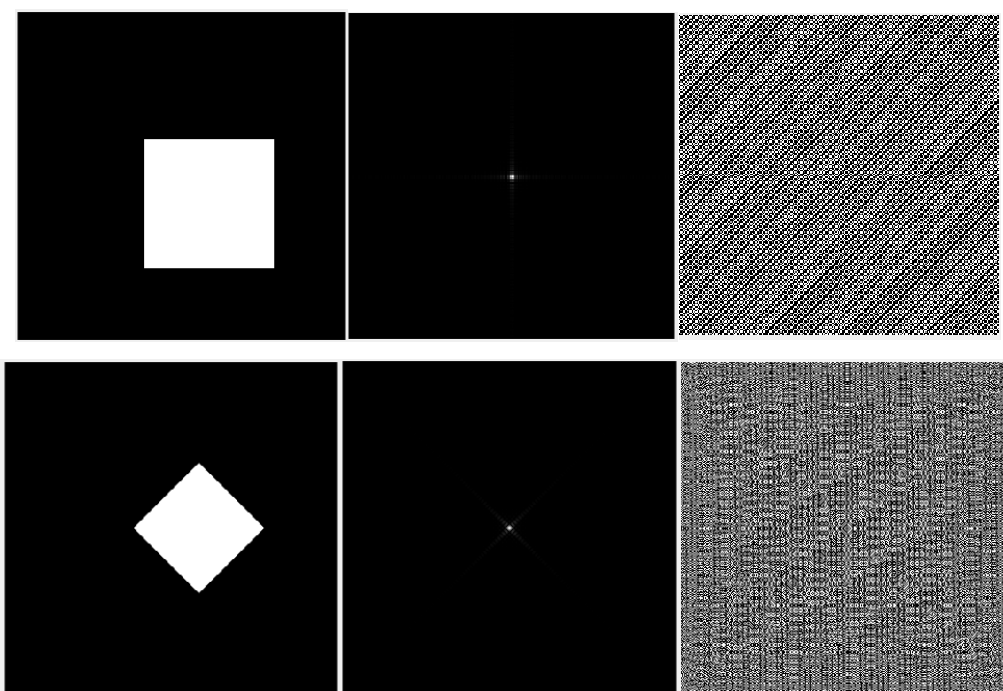
不同角度的矩形图像傅里叶变换代码如下：

```

f2 = imrotate(f1,45,'bilinear'); %rotate
figure(3);
subplot(2,3,1),imshow(f1,[]);
subplot(2,3,2), imshow(abs(f1_DFT),[]); %show the margin
subplot(2,3,3), imshow(angle(f1_DFT*180/pi)); %show the frequency
f2_DFT = fftn(f2);
f2_DFT = fftshift(f2_DFT);
subplot(2,3,4),imshow(f2,[]);
subplot(2,3,5), imshow(abs(f2_DFT),[]); %show the margin
subplot(2,3,6), imshow(angle(f2_DFT*180/pi)); %show the frequency

```

结果如下：



上图中，左图是空域中的原图像，中图是频谱图，右图是相位图。可以看出在频谱图可以反应出形状的方向。但相位图相对来说难以理解。