项目编号Project 04-01~04-05&谱旋转性质

学生姓名：郝裕玮&张闯

截止日期：10.19

上交日期：10.19

摘要

本次实验共完成了6个项目：实现二维快速傅里叶变换，研究傅里叶谱相关性质（绘制图像，探究图像旋转），实现多种滤波器并探究频域相关问题。

**小组分工：**

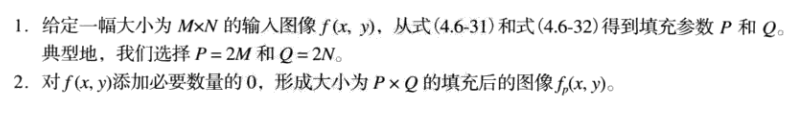
**郝裕玮：Project 04-01~04-03**

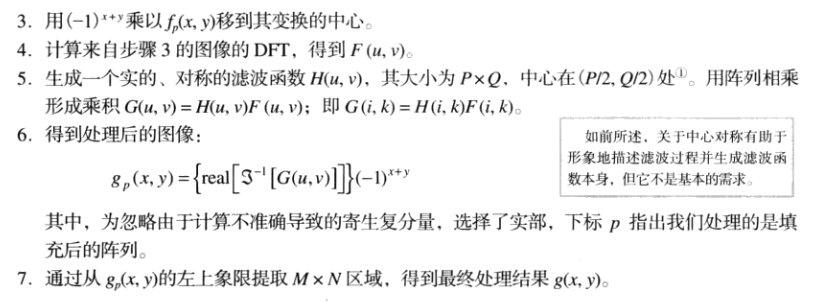
**张闯：Project 04-04~04-05 & 研究谱旋转性质**

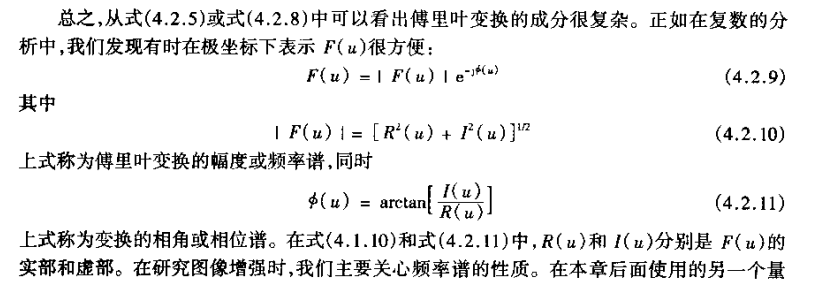
**Project 04-01**

（1）技术讨论：

代码具体思路为：根据如下图所示的步骤进行代码编写即可。









代码如下所示（具体分析已包含在代码注释中）：

% 读取图像

image=imread('C:\Users\93508\Desktop\1.jpg');

image=im2double(image); %图像处理需要双精度

figure(1);

imshow(image,[]);

title('初始图像');

% 扩充原图矩阵得到fp(x,y)

[M,N]=size(image);

P=2\*M;

Q=2\*N;

image\_fp=zeros(P,Q);

image\_fp(1:M,1:N)=image(1:M,1:N);

% 用(-1)^(x+y)乘以fp(x,y)，将其移动到变换中心

for x=1:P

    for y=1:Q

        image\_fp(x,y)=image\_fp(x,y).\*(-1)^(x+y);

    end

end

% 对图像做二维傅里叶变换

image\_F=fft2(image\_fp);

% 生成理想低通滤波器H(u,v)

image\_H=zeros(P,Q);

D0=60;

for x=1:P

    for y=1:Q

        D=sqrt((x-M)^2+(y-N)^2);

        if D>D0

            image\_H(x,y)=0;

        else

            image\_H(x,y)=1;

        end

    end

end

figure(2);

imshow(image\_H);

title('理想低通滤波器');

% 阵列相乘得到乘积G(u,v)=H(u,v)\*F(u,v)

image\_G=image\_H.\*image\_F;

% 求G的IDFT并取其实部，再乘以(-1)^(x+y),得到处理后的图像gp(x,y)

image\_gp=real(ifft2(image\_G));

for x=1:P

    for y=1:Q

        image\_gp(x,y)=image\_gp(x,y).\*(-1)^(x+y);

    end

end

% 从gp(x,y)的左上象限提取M\*N区域得到最终处理结果g(x,y)

image\_g=image\_gp(1:M,1:N);

figure(3);

imshow(image\_g,[]);

title('最终结果');

% 画出傅里叶频谱图

R=real(image\_F); %提取实部

I=imag(image\_F); %提取虚部

image\_F1=log(1+abs(image\_F));

figure(4);

imshow(image\_F1,[]);

title('傅里叶变换幅度谱');

figure(5);

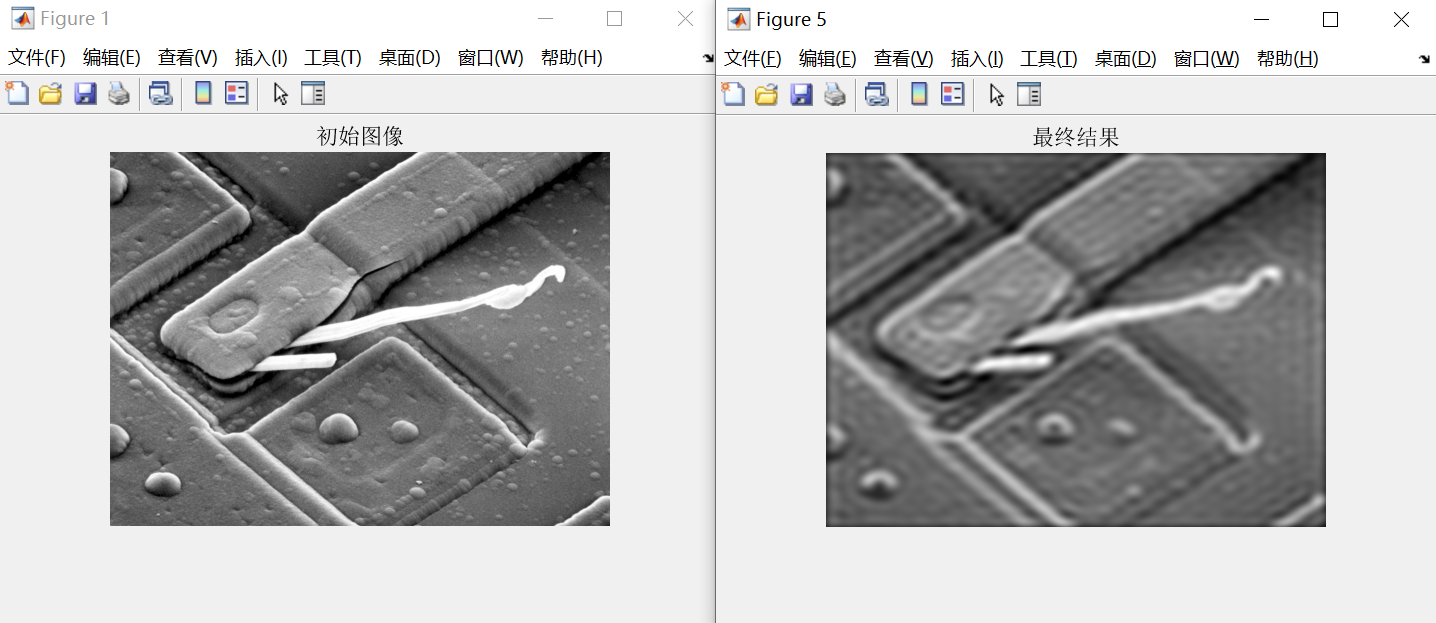
image\_F2=atan(I./R);

imshow(image\_F2,[]);

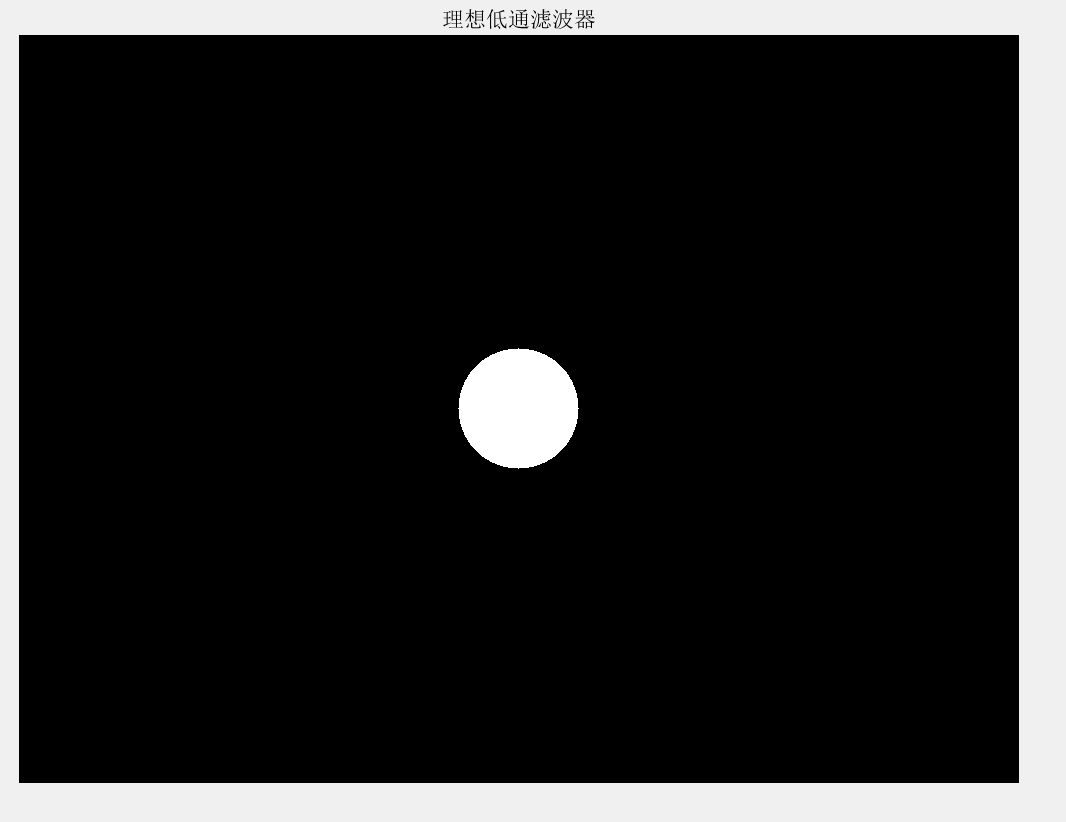
title('傅里叶变换相位谱');

（2）结果讨论：

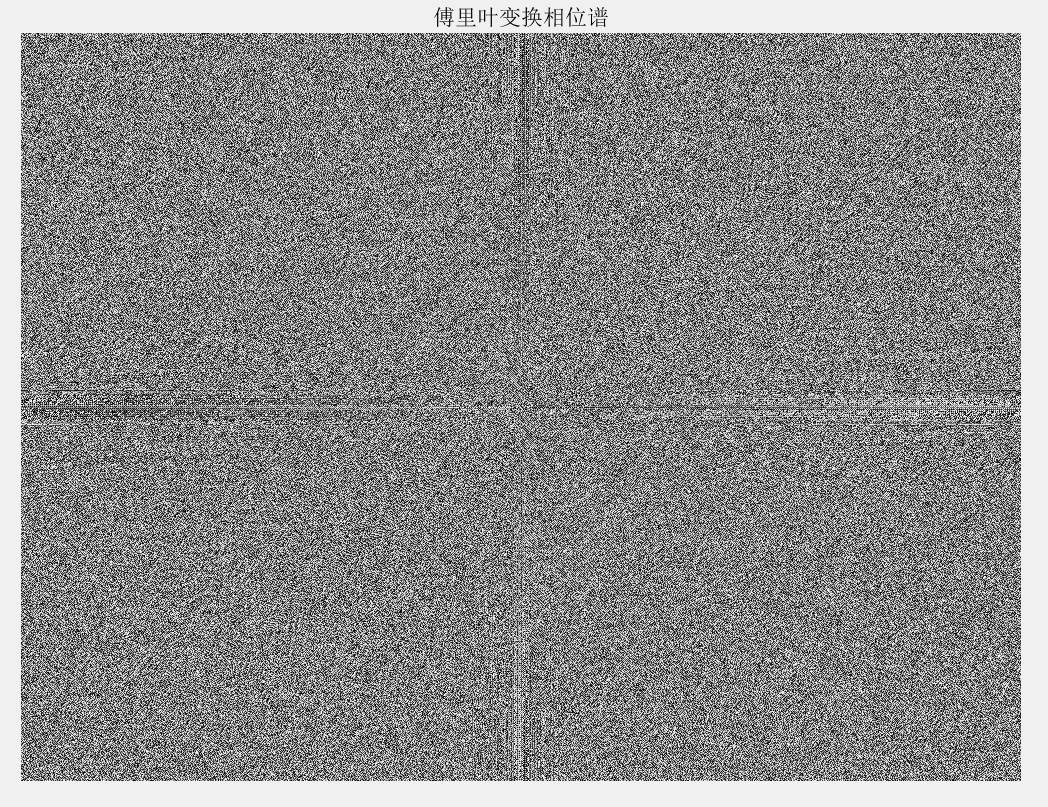
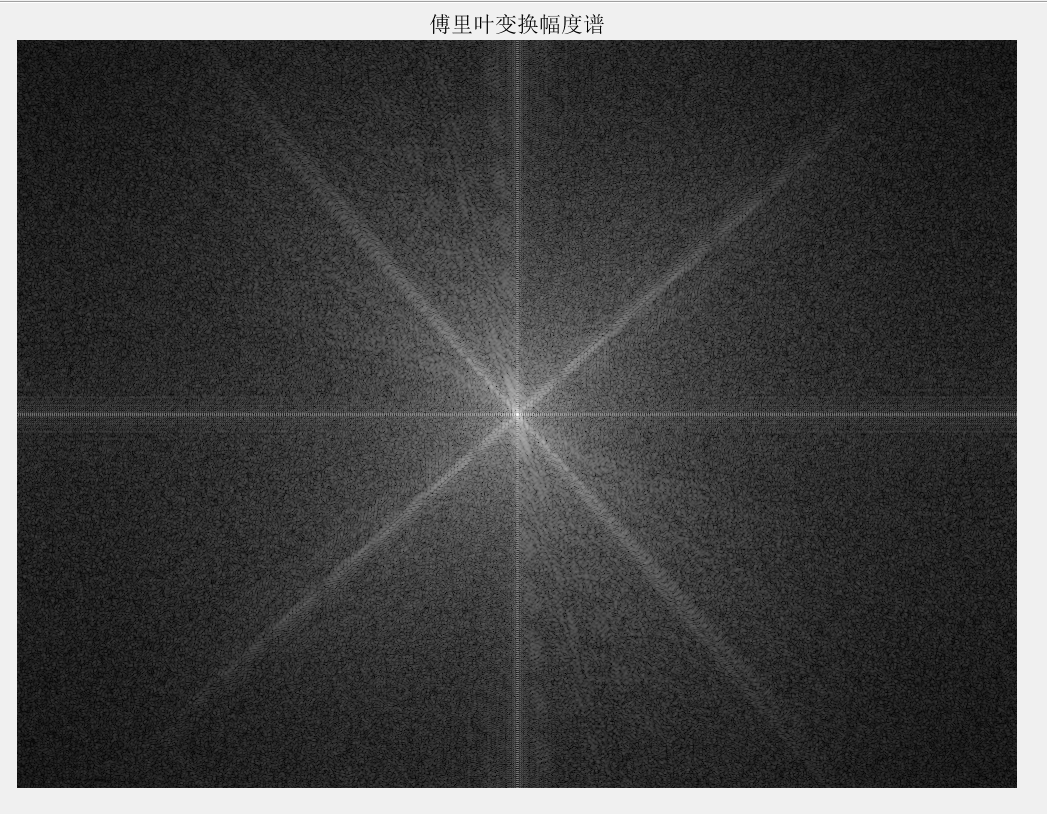
初始图像和最终结果的对比（见下页）：



理想低通滤波器（=60）：



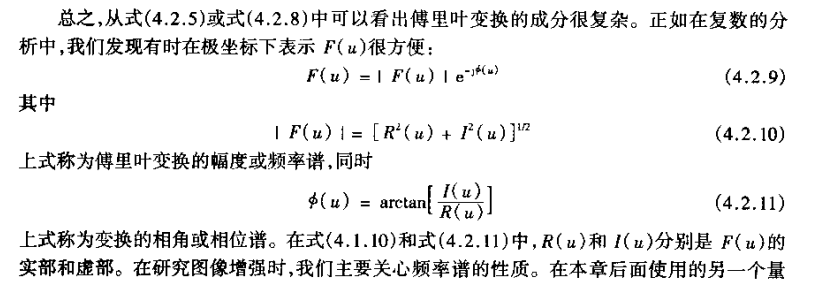
傅里叶频谱图为：



**Project 04-02**

（1）技术讨论：

代码大多数部分和04-01一致，代码如下所示（具体分析详见代码注释）：



% 读取图像并计算图像的平均值

image=imread('C:\Users\93508\Desktop\2.jpg');

avg=mean2(image);

disp(['图像的平均值为',num2str(avg)]);

%图像处理需要双精度

image=im2double(image);

figure(1);

imshow(image,[]);

title('初始图像');

% 扩充原图矩阵得到fp(x,y)

[M,N]=size(image);

P=2\*M;

Q=2\*N;

image\_fp=zeros(P,Q);

image\_fp(1:M,1:N)=image(1:M,1:N);

% 用(-1)^(x+y)乘以fp(x,y)，将其移动到变换中心

for i=1:P

    for j=1:Q

        image\_fp(i,j)=image\_fp(i,j)\*(-1)^(i+j); %平移函数，移到其变换的中心

    end

end

% 对图像做二维傅里叶变换

image\_F=fft2(image\_fp);

% 画出傅里叶频谱图

X=real(image\_F); %提取实部

Y=imag(image\_F); %提取虚部

image\_F1=log(1+abs(image\_F));

figure(2);

imshow(image\_F1,[]);

title('傅里叶变换幅度谱');

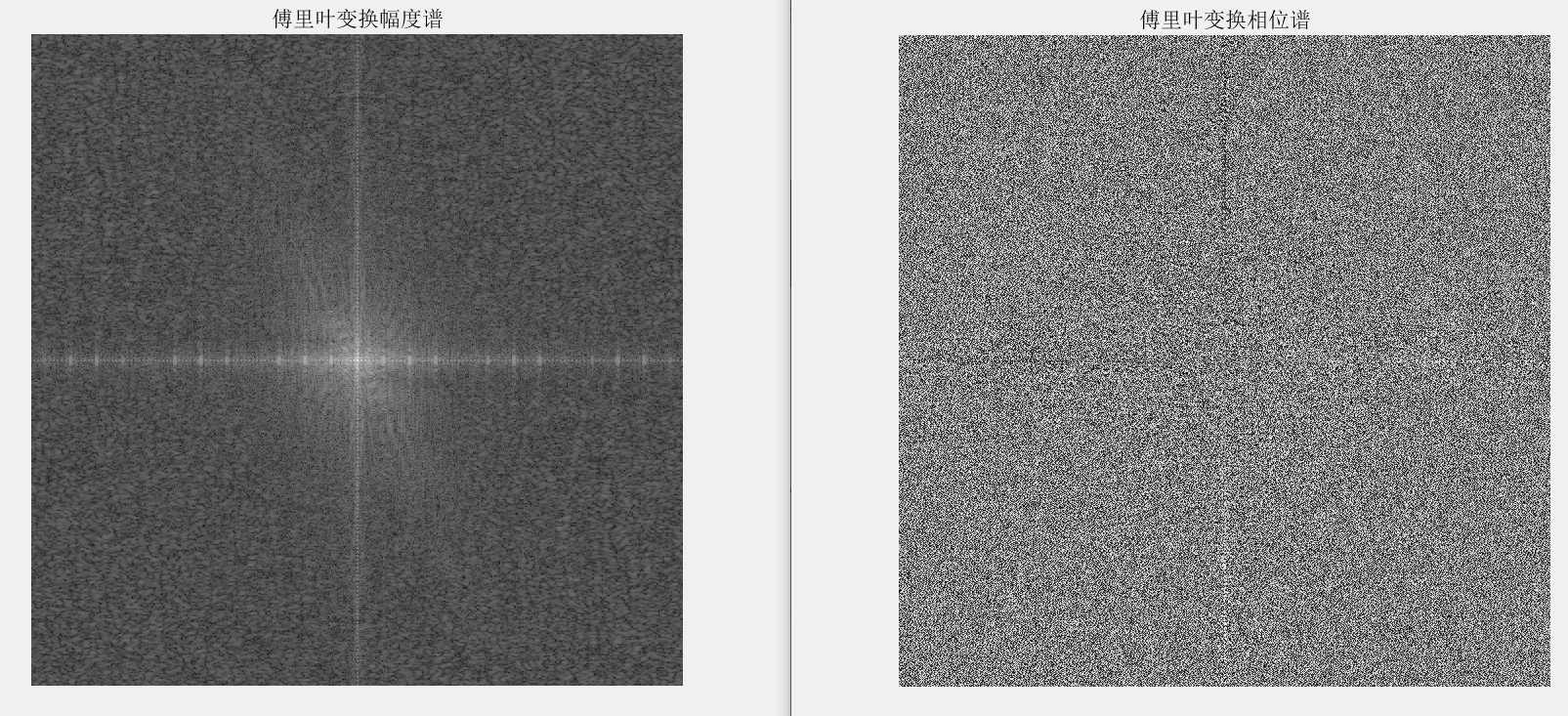
figure(3);

image\_F2=atan(Y./X);

imshow(image\_F2,[]);

title('傅里叶变换相位谱');

（2）结果讨论：



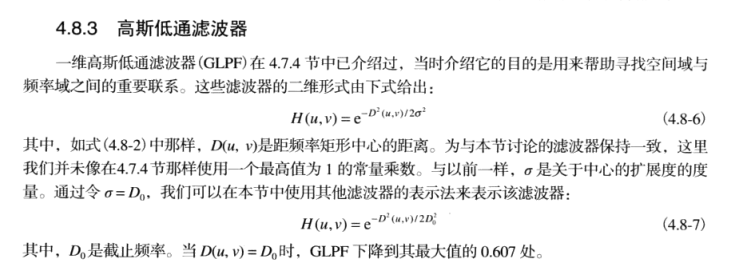
图像的平均值为：207.3635



**Project 04-03**

（1）技术讨论：

代码大多数部分和04-01一致，代码如下所示（具体分析详见代码注释）：



% 读取图像

image=imread('C:\Users\93508\Desktop\2.jpg');

image=im2double(image);

figure(1);

imshow(image);

title('初始图像');

% 扩充原图矩阵得到fp(x,y)

[M,N]=size(image);

P=2\*M;

Q=2\*N;

image\_fp=zeros(P,Q);

image\_fp(1:M,1:N)=image(1:M,1:N);

% 用(-1)^(x+y)乘以fp(x,y)，将其移动到变换中心

for x=1:P

    for y=1:Q

        image\_fp(x,y)=image\_fp(x,y).\*(-1)^(x+y);

    end

end

% 对图像做二维傅里叶变换

image\_F = fft2(image\_fp);

% 高斯低通滤波器

image\_H=zeros(P,Q);

P=2\*M;

Q=2\*N;

D0=30;

for i=1:P

    for j=1:Q

        image\_H(i,j)=exp(-((i-P/2)^2+(j-Q/2)^2)/(2\*D0^2));

    end

end

figure(2);

imshow(image\_H);

title('高斯低通滤波器');

% 阵列相乘得到乘积G(u,v)=H(u,v)\*F(u,v)

image\_G = image\_F .\* image\_H;

% 求G的IDFT并取其实部，再乘以(-1)^(x+y),得到处理后的图像gp(x,y)

image\_gp = real(ifft2(image\_G));

for x=1:P

    for y=1:Q

        image\_gp(x, y)=image\_gp(x,y).\*(-1)^(x+y);

    end

end

% 从gp(x,y)的左上象限提取M\*N区域得到最终处理结果g(x,y)

image\_g=image\_gp(1:M, 1:N);

figure(3);

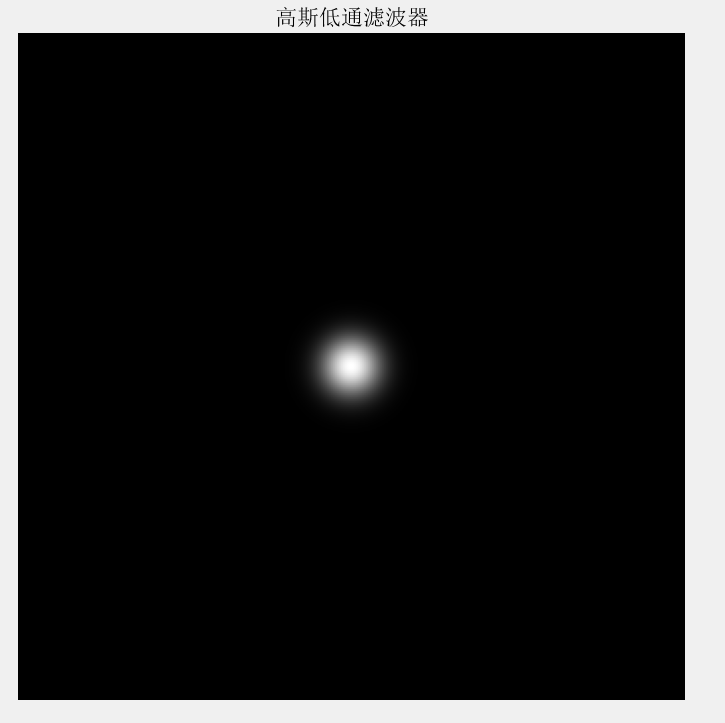
imshow(image\_g, []);

title('高斯低通滤波后的图像');

（2）结果讨论：

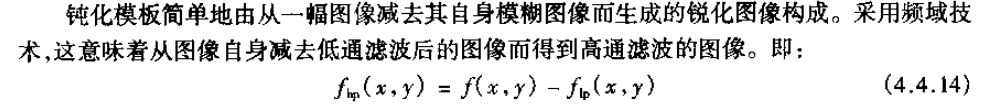


高斯低通滤波器（截止频率=30）：



**Project 04-04**

（1）技术讨论：



%读入原图像

img = imread('Fig4.11(a).jpg');

img = im2double(img);

figure(1);

imshow(img);

title('原图像');

% 扩充原图像

[M, N] = size(img);

P = 2 \* M;

Q = 2 \* N;

img\_fp = zeros(P, Q);

img\_fp(1:M, 1:N) = img(1:M, 1:N);

% 计算滤波函数;

alf = 80;

H = zeros(P, Q);

for i = 1:P

    for j = 1:Q

        H(i, j) = exp(-((i-P/2)^2 + (j-Q/2)^2) / (2 \* alf^2));

    end

end

figure(2);

imshow(H);

title('滤波函数');

%将f移到变换重心

img\_f = zeros(P, Q);

for x = 1:P

    for y = 1:Q

        img\_f(x, y) = img\_fp(x, y) .\* (-1)^(x+y);

    end

end

%计算图像的傅里叶变换

img\_F = fft2(img\_f);

%用阵列相乘得到img\_G，再用傅里叶反变换得到img\_g

img\_G = img\_F .\* H;

img\_g = real(ifft2(img\_G));

%将f变换回左上角，得到处理后的图像

for x = 1:P

    for y = 1:Q

        img\_g(x, y) = img\_g(x, y) .\* (-1)^(x+y);

    end

end

%截取左上角部分，得到高斯低通滤波图像

img\_o = img\_g(1:M, 1:N);

%用原图像减去高斯低通滤波图像，得到高斯高通滤波图像

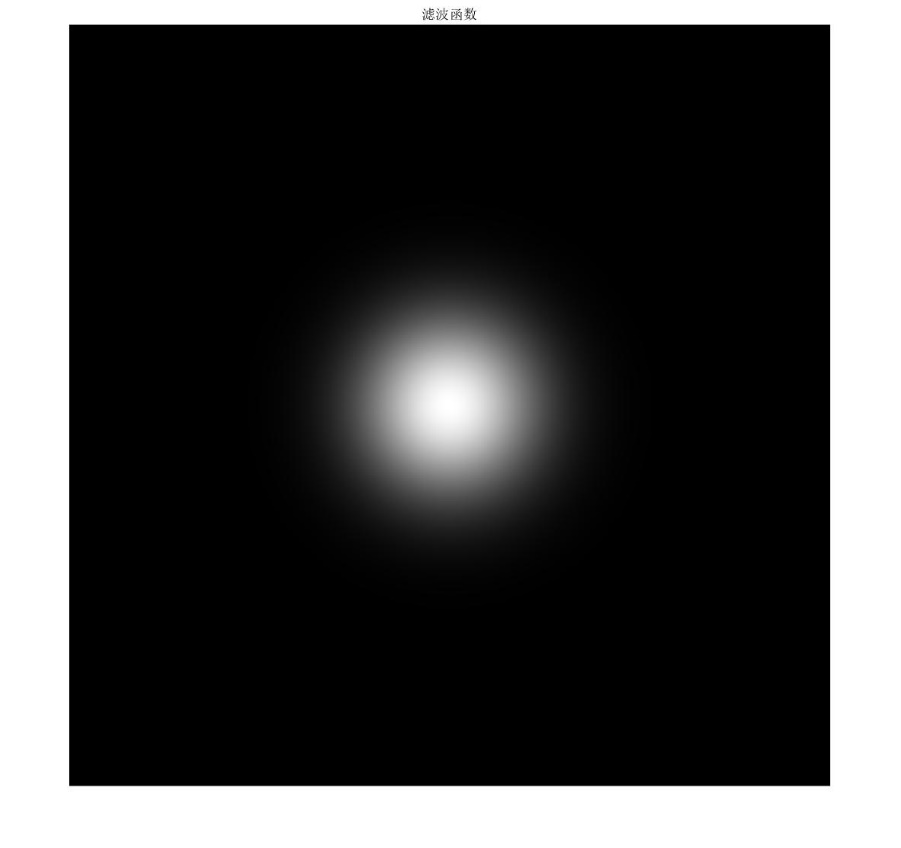
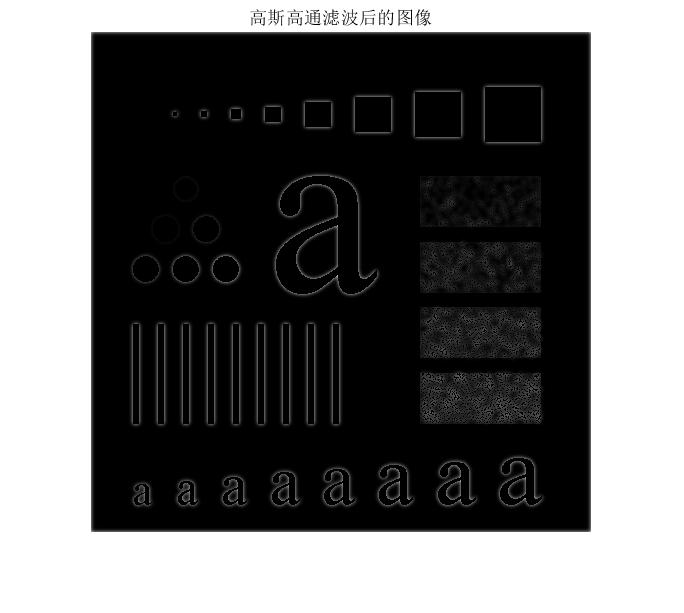
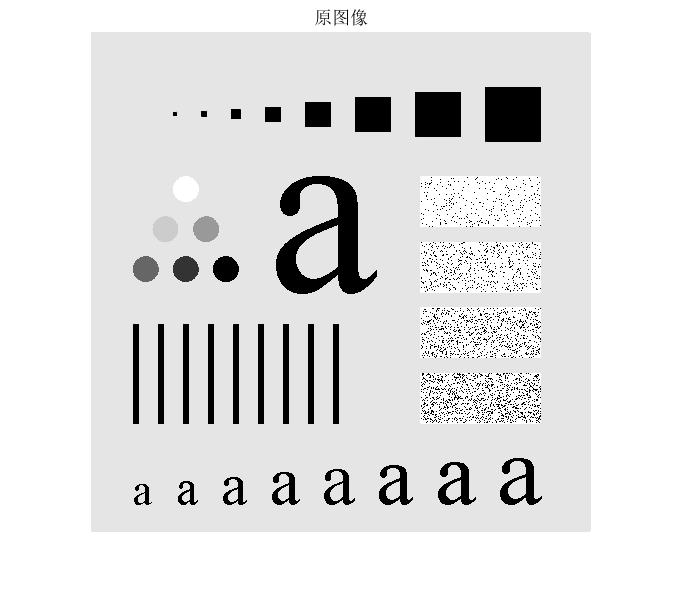
img\_4=img-img\_o;

figure(3);

imshow(img\_4);

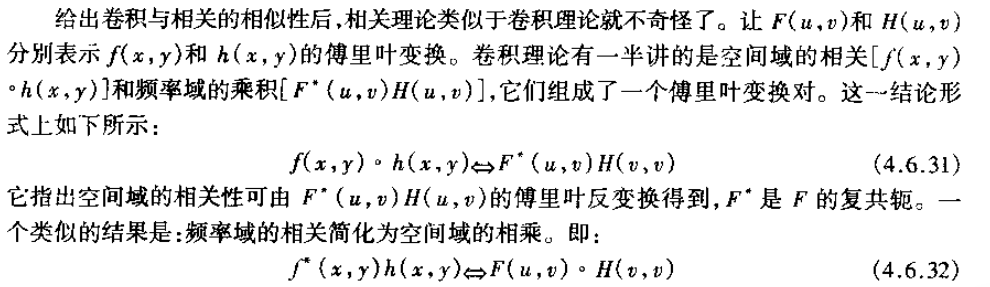
title('高斯高通滤波后的图像')

（2）结果讨论：



**Project 04-05**

（1）技术讨论：



% 读入原图像

imga = imread('Fig4.41(a).jpg');

imga = im2double(imga);

figure(1);

imshow(imga);

title('原图像');

% 读入模板图像

imgb = imread('Fig4.41(b).jpg');

imgb = im2double(imgb);

figure(2);

imshow(imgb);

title('模板图像');

[A,B]=size(imga);

[C,D]=size(imgb);

% 扩充图像，取P>=A+C,Q>+B+D，选择298\*298的相等延拓尺度

P=298;

Q=298;

f=zeros(P,Q);

g=zeros(P,Q);

f(1:A,1:B)=imga(1:A,1:B);

figure(3);

imshow(f);

title('延拓原图像');

g(1:C,1:D)=imgb(1:C,1:D);

figure(4);

imshow(g);

title('延拓模板图像');

for x=1:P

    for y=1:Q

        f(x,y)=f(x,y).\*(-1)^(x+y);

        g(x,y)=g(x,y).\*(-1)^(x+y);

    end

end

% 利用相关理论对图像进行图像相关处理

F=fft2(f);

G=fft2(g);

Img=F.\*conj(G);

img=ifft2(Img);

for x=1:P

    for y=1:Q

        img(x,y)=img(x,y).\*(-1)^(x+y);

    end

end

img=real(img);

img = mat2gray(img);

figure(5);

imshow(img);

title('图像相关结果');

% 寻找最大值及最大值坐标

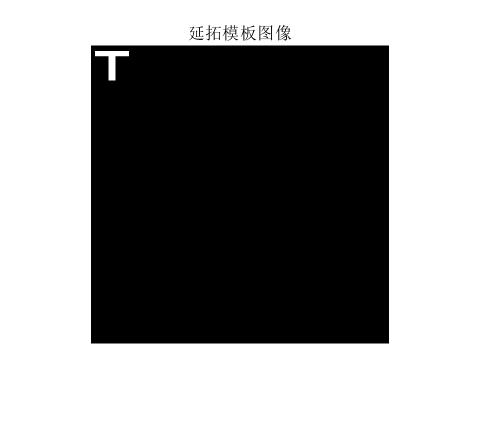
max\_value = max(max(img));

[row,col] = find(img == max\_value);

disp(['row: ', num2str(row), ' col: ', num2str(col)]);

（2）结果讨论：

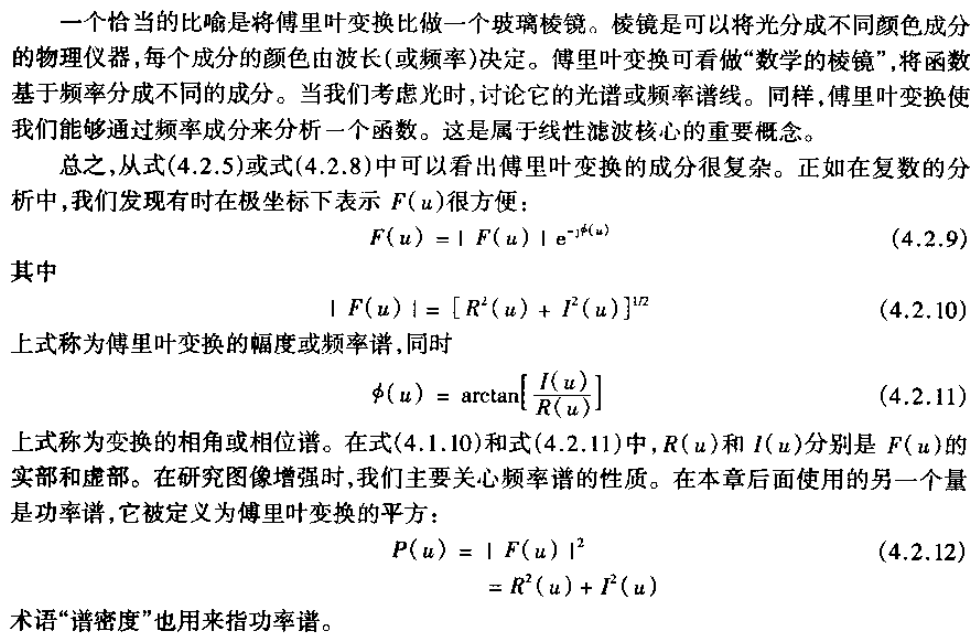






**研究谱旋转性质**

（1）技术讨论：



image=imread('Fig4.04(a).jpg');

image=im2double(image);

%angle可设置为别的角度

angle=0;

image=imrotate(image,angle);

figure(1);

imshow(image);

title('旋转0度后的图像');

% 对图像进行扩充

[M,N]=size(image);

M=2\*M;

N=2\*N;

Image=zeros(M,N);

Image(1:M/2,1:N/2)=image(1:M/2,1:N/2);

% 将图像移动到变换中心

for x=1:M

    for y=1:N

        Image(x,y)=Image(x,y).\*(-1)^(x+y);

    end

end

Image=Image(1:M/2,1:N/2);

%对图像进行傅里叶变换，计算出频谱图像

T=fft2(Image);

% 对图像取对数，增强效果

F=log(1+abs(T));

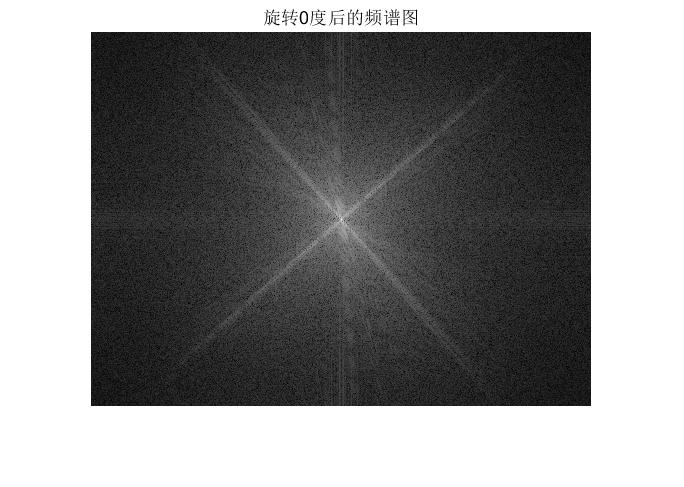
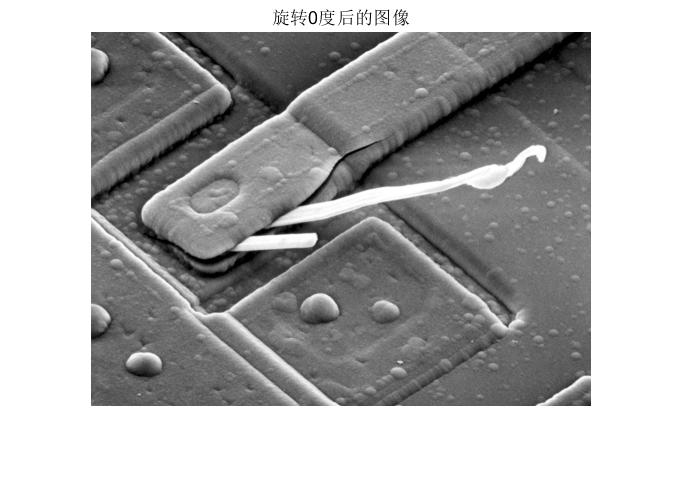
F=mat2gray(F);

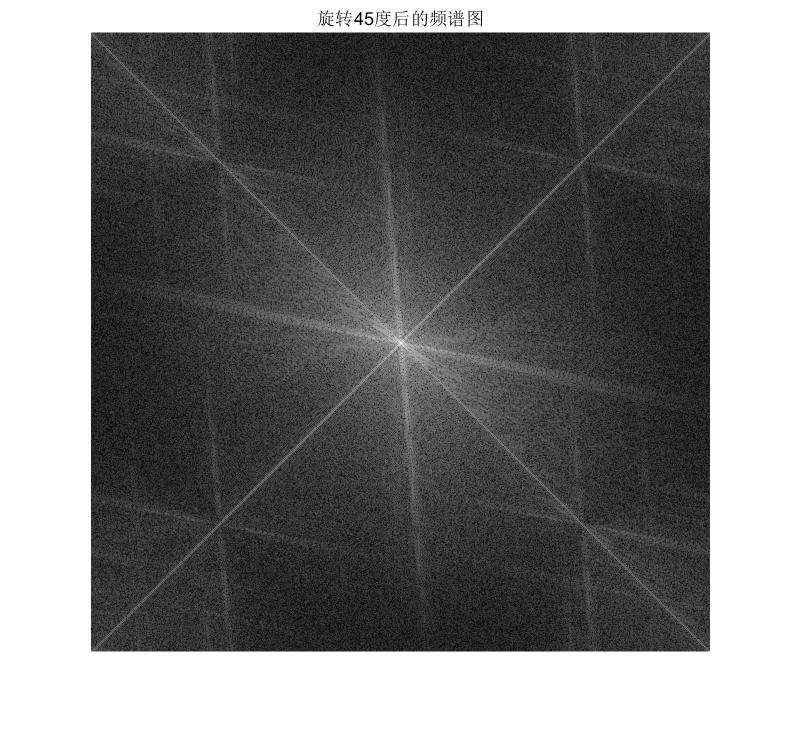
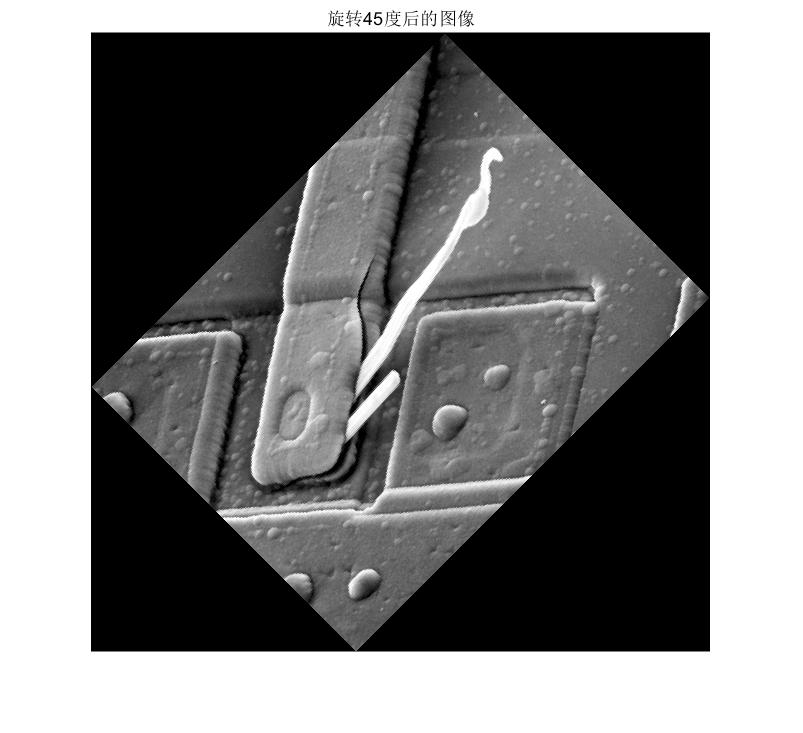
figure(2);

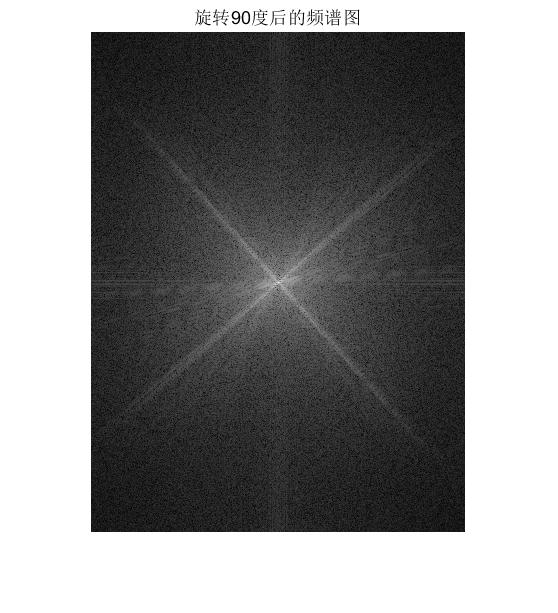
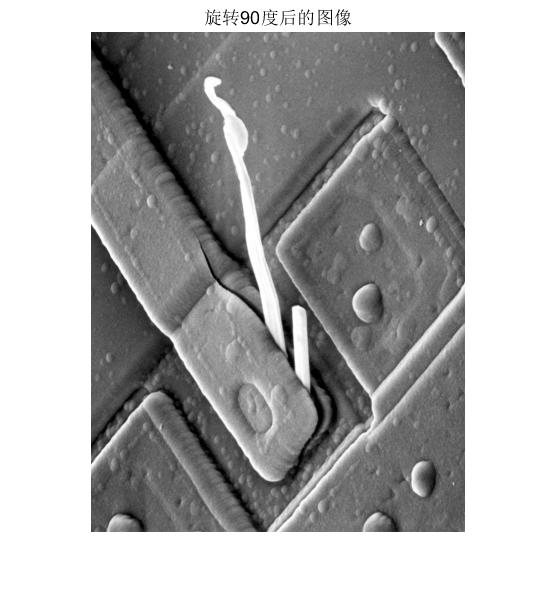
imshow(F);

title('频谱图');

（2）结果讨论（见下页）：

****

****

****