大作业报告：4f系统实现边缘提取

PB18020539黄韫飞

基于傅里叶光学中的4f系统（所有系统参数自定），实现光学图像的边缘提取。研究：1）理论推导出边缘提取算子尺寸与空间复滤波器间空间分布的关系，可利用严格的公式进行推导；2）给出空间复滤波器的振幅和位相分布；3）找一些图片，验证滤波器在边缘提取的效果。

第一部分 边缘提取算子对应空间复滤波器的理论推导

假设算子对应的矩阵为，要变换到N×N的屏幕上，则算子的(1,1)点处在屏幕的位置，则采用离散傅里叶变换：

其中。X即为空间复滤波器的矩阵求和范围缩小至算子所在的区域，则：

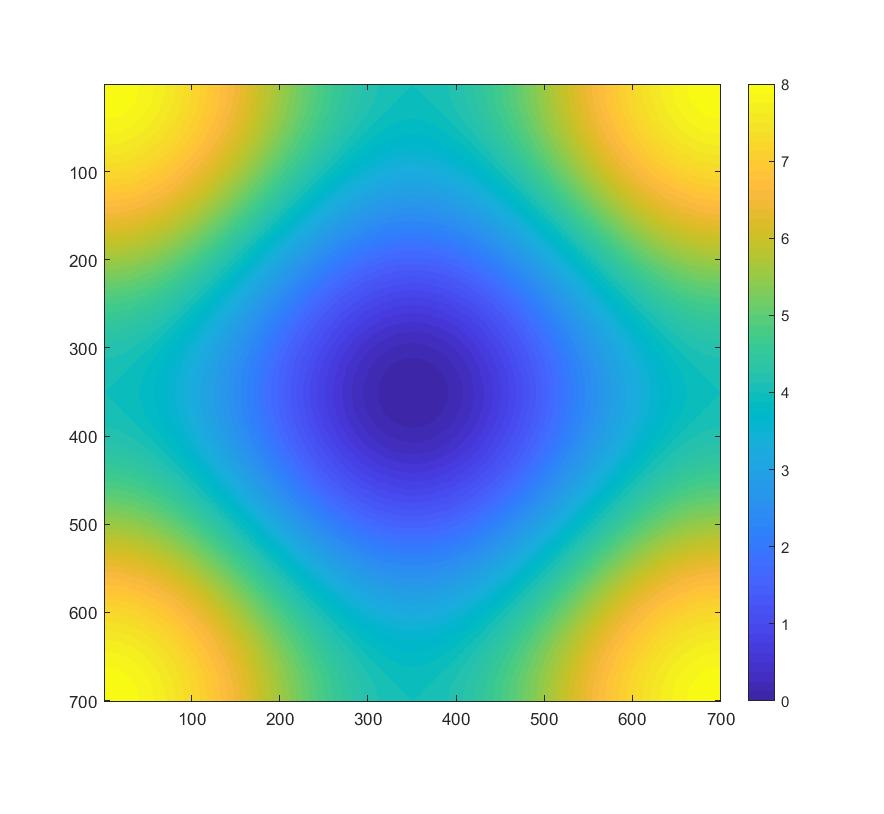
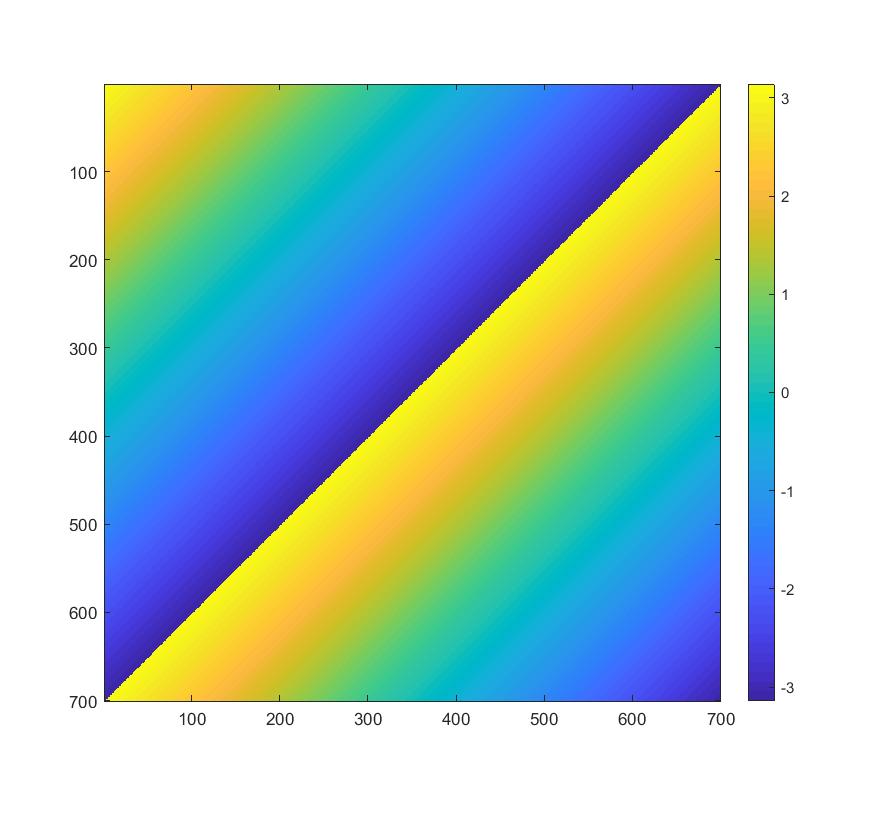
其中。

第二部分 空间复滤波器的振幅和位相分布

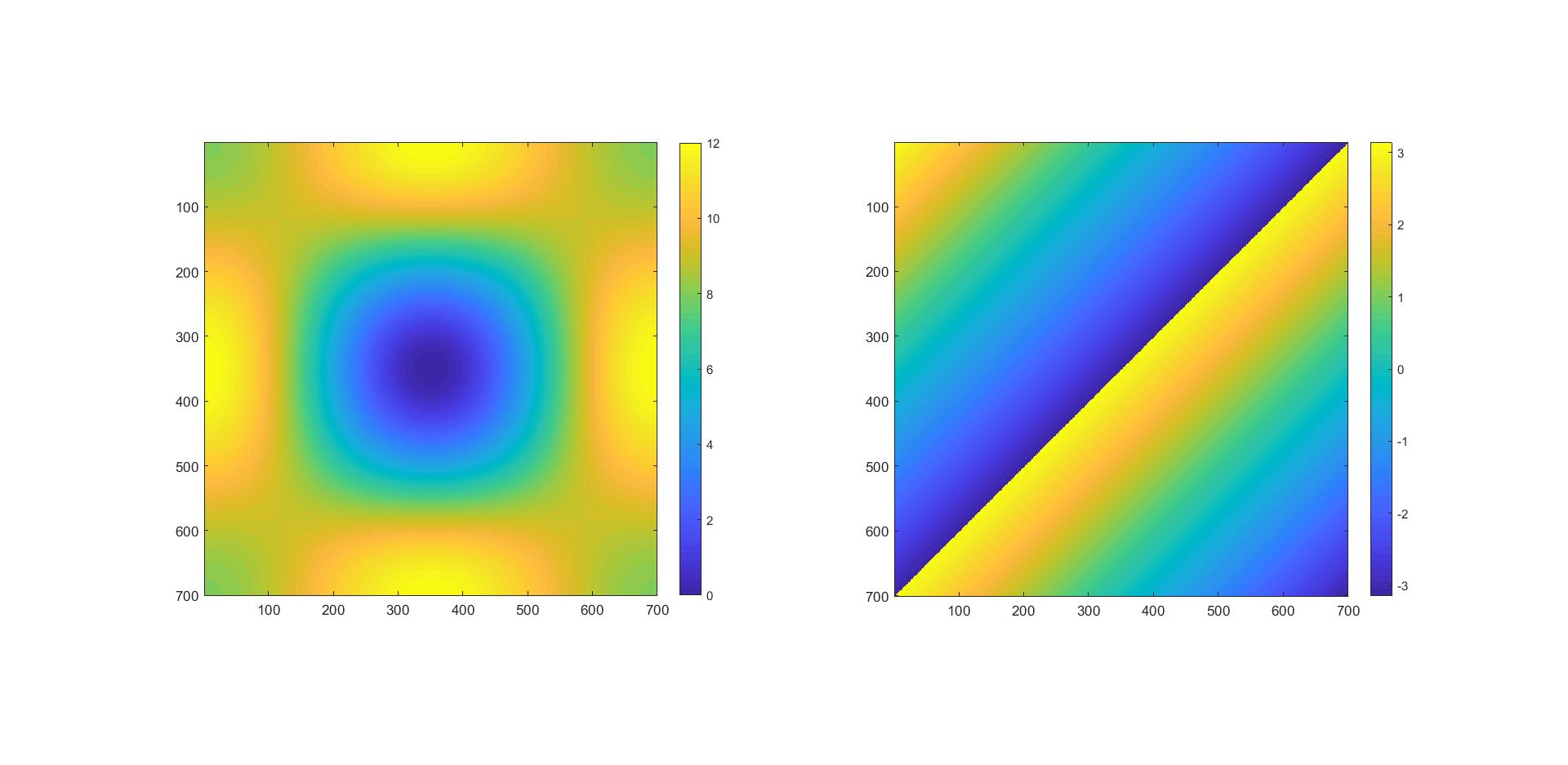
本次程序采用的是Laplace算子。写成差分形式为：

写成算符的形式为：

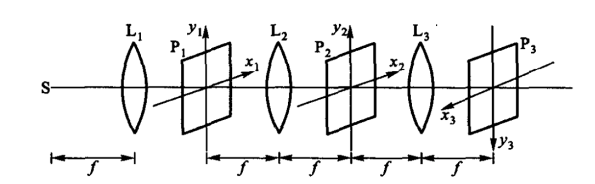
做FFT,并将FFT得到的H矩阵设置为与屏幕一样大的N×N，文件夹中USTC.jpg的宽度为N=700，用到的函数为fftshift(fft2(A,N,N))，运行得到H的振幅和位相分布：



在一些文献中，也用拉普拉斯算子的另一种形式：

其傅里叶变换后得到的振幅和位相如下：

第三部分 程序简介

下图是传播的光路

P1为待处理的图像g(x1,y1)。用RSDiff.m函数模拟衍射传播f后，至L2前入射面。得到U1。经过透镜相当于乘相位。之后传播f至P2面，P2面为g的频谱G(u,v)。与第二部分中得到的H相乘得到G(u,v)H(u,v)。再经过两段f和一个透镜，相当于傅里叶变换，得到g(x3,y3)=g\*h，\*代表卷积。

第四部分 程序与运行结果

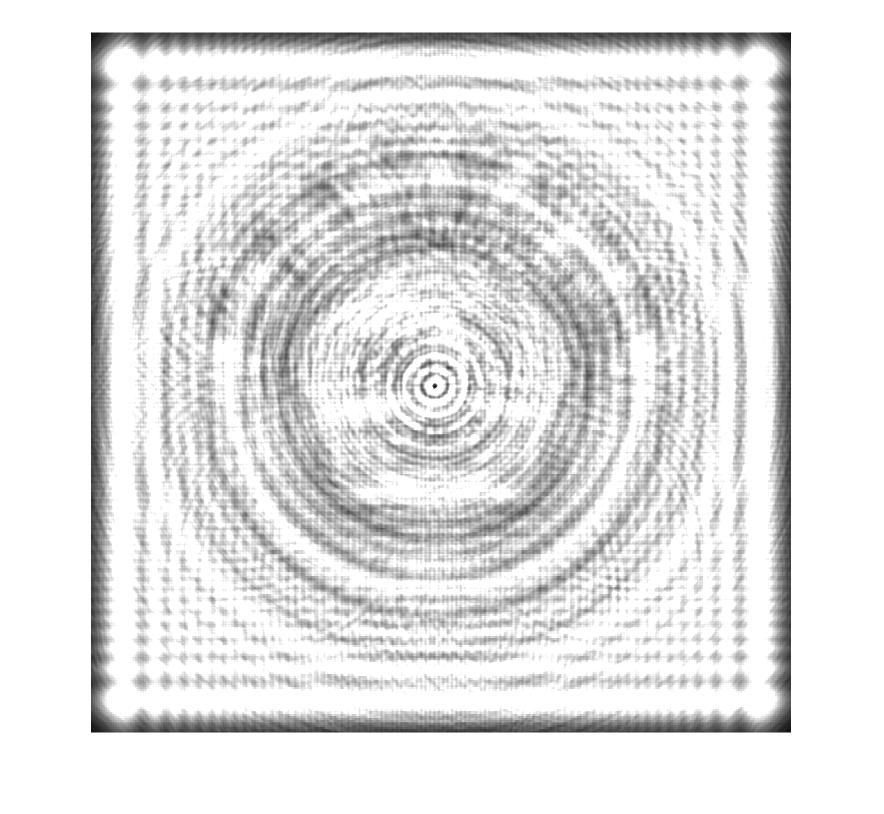
|  |
| --- |
| 1. % 4f system in fourier optics to detect the edge of a picture 2. clear,clc,clf 3. Figure=im2double(imread('USTC.jpg')); 4. Input(:,:)=sqrt(Figure(:,:,3)); 5. nfig=length(Input); 6. lambda=1;f=1500;stepxy=1;k=2\*pi/lambda;nscreen=nfig; 7. A=[0,1,0;1,-4,1;0,1,0]; 8. H=fftshift(fft2(A,nscreen,nscreen)); 9. x=-(nscreen/2):stepxy:(nscreen/2-stepxy);y=x'; 10. [XX,YY]=meshgrid(x,y); 11. r=sqrt(XX.^2+YY.^2); 12. Screen=zeros(nscreen,nscreen); 13. Screen((nscreen-nfig)/2+1:(nscreen+nfig)/2,(nscreen-nfig)/2+1:(nscreen+nfig)/2)=Input; 14. U1=RSDiff(f,x,k,Screen); 15. Phi=exp(-1i\*k\*r.^2/(2\*f)); 16. imshow(abs(U1).^2) 17. U11=U1.\*Phi; 18. U2=RSDiff(f,x,k,U11); 19. figure; 20. imshow(abs(U2).^2); 21. U22=U2.\*H; 22. figure; 23. imshow(abs(U22).^2) 24. U23=RSDiff(f,x,k,U22); 25. U24=U23.\*Phi; 26. U3=RSDiff(f,x,k,U24); 27. figure; 28. imshow(abs(U3).^2); 29. figure; 30. imshow(abs(conv2(Input,A,'same')).^2); |

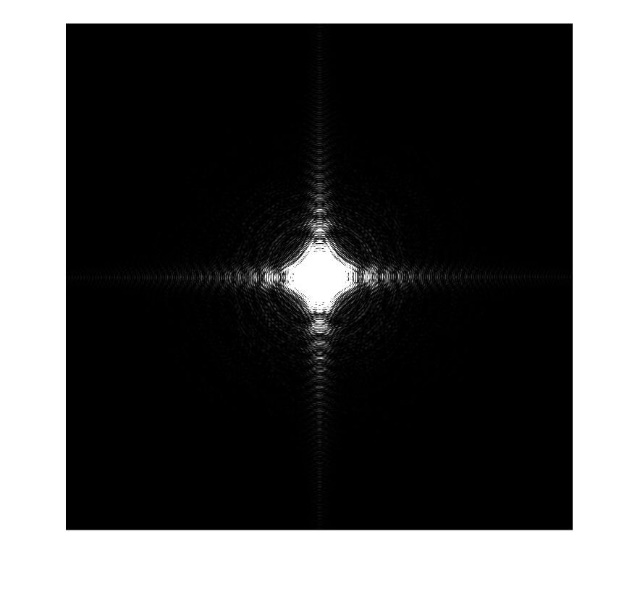
其中RSDiff.m函数如下：

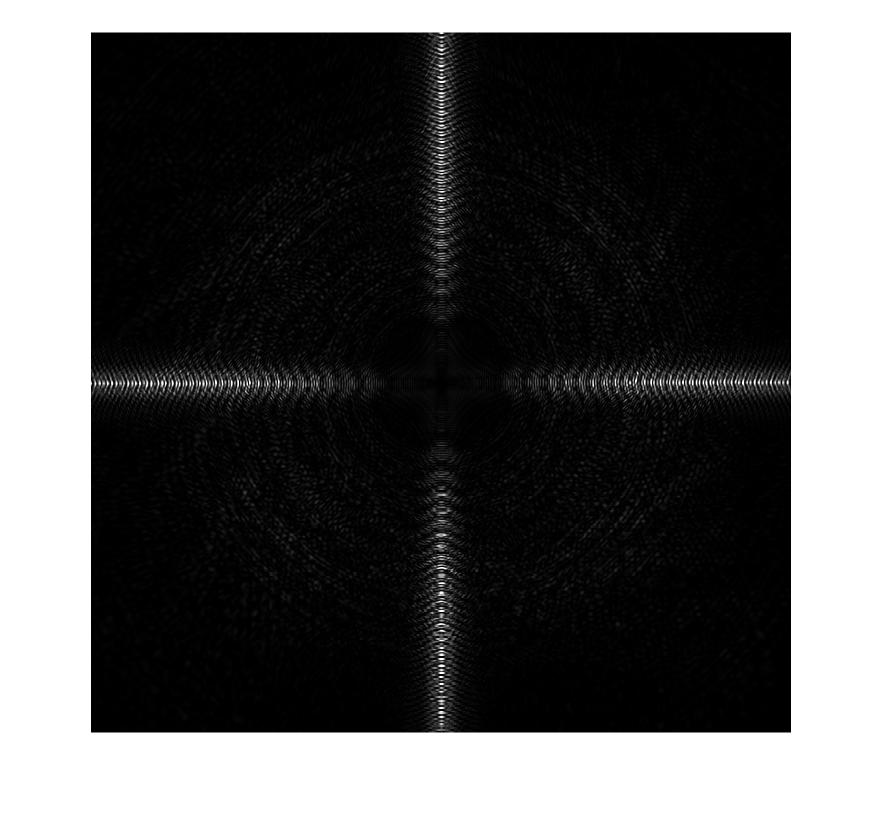
|  |
| --- |
| 1. function Out = RSDiff(z,s,k,object) 2. %function RSDiff uses the formula in article2006 to calculate image 3. eta=s;x=s;y=s; 4. n=length(s); 5. U=[object,zeros(n,n-1);zeros(n-1,n),zeros(n-1,n-1)]; 6. X=[x(1)-s(n+1-(1:(n-1))),x((n:2\*n-1)-n+1)-s(1)]; 7. Y=[y(1)-eta(n+1-(1:(n-1))),y((n:2\*n-1)-n+1)-eta(1)]; 8. [XX,YY]=meshgrid(X,Y); 9. r=sqrt(XX.^2+YY.^2+z^2); 10. G=1/(2\*pi)\*exp(1i\*k\*r)./r\*z./r.\*(1./r-1i\*k); 11. S=ifft2(fft2(U).\*fft2(G)); 12. Out=S(n:end,n:end); 13. end |

运行结果：

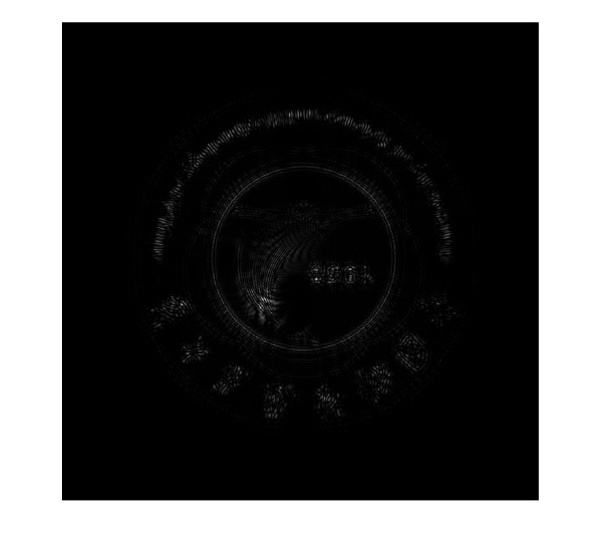
待处理的图像（大小为700×700像素）：

传播至U1的图像：

在透镜后焦平面的成像（即上图的频谱）

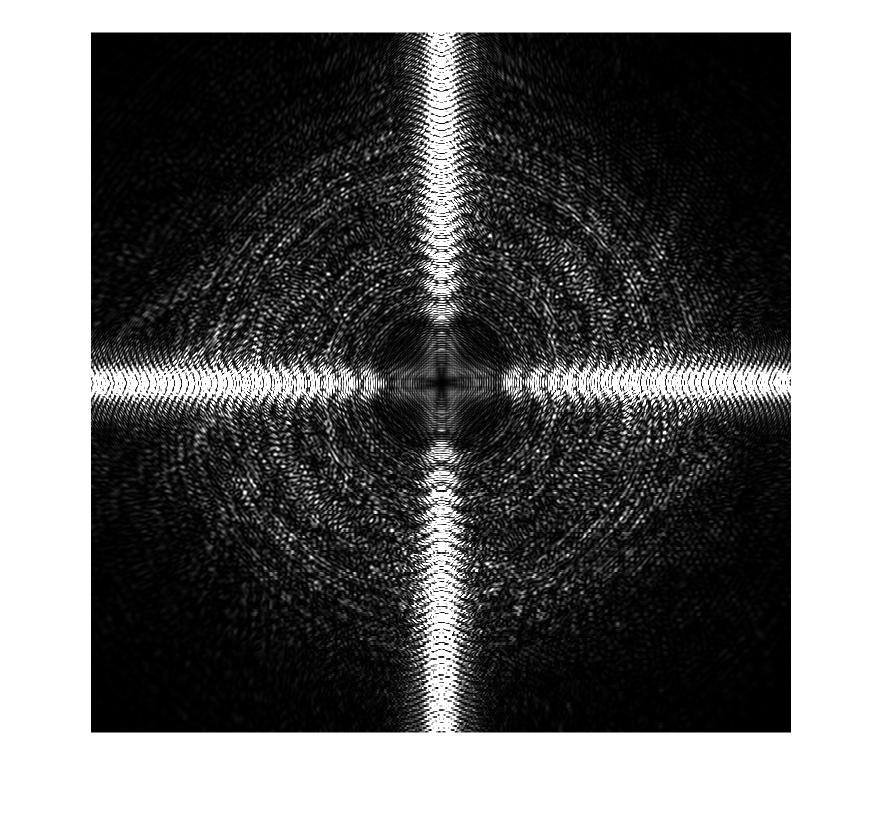
与FFT变换后的边缘提取算子相乘，得到：

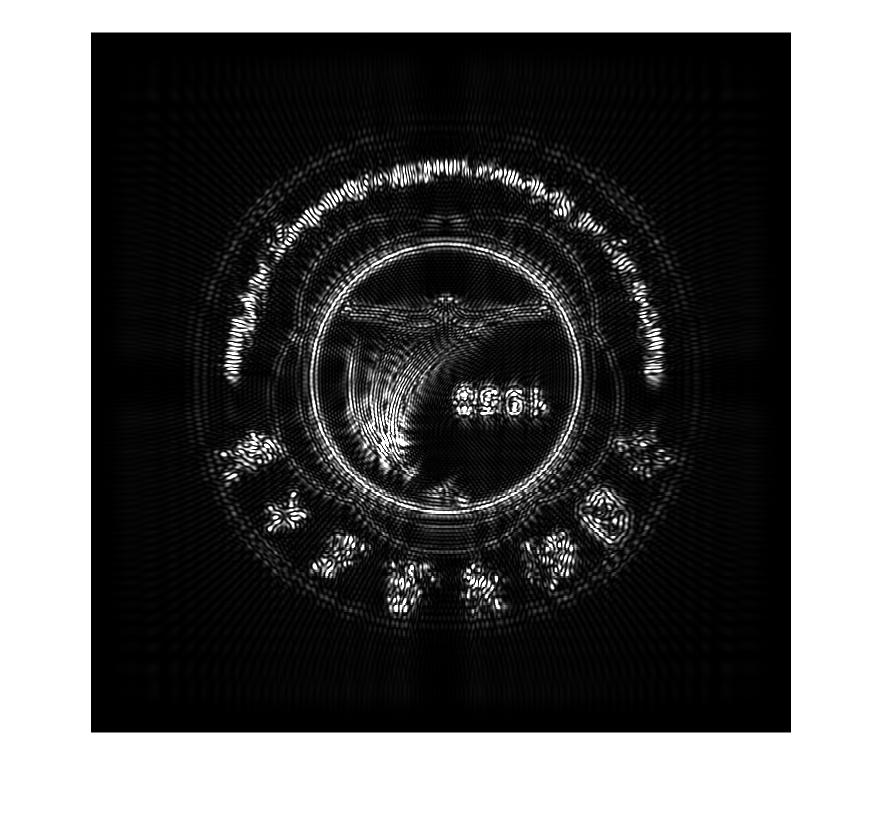
再经过两段f和一个透镜，得到原图倒立的边缘像：



与下图直接用conv2函数卷积相比：

如果改用第二种形式的Laplace算子矩阵，即：

传播至U1，U2的图像与原来相同，乘新的算子后得到：

继续传输到L3的后焦平面：

直接利用算符进行卷积得到：

相比较而言，由于衍射效应，4f系统得到的像的边缘分辨不如直接从数学上卷积清晰。但是仍然可以清楚看到圆圈、五瓣梅花、数字1958、文字等元素的边缘。考虑到本次只是简单探究，我并没有进一步尝试调整参数使得像更清晰。如果要调整参数，需要考虑衍射斑和物的大小关系以及采样频率和空间频率的关系。