大作业报告：4f 系统实现边缘提取

PB18020539 黄镭飞

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$A_{m\times m}$，要变换到 N×N 的屏幕上，则算子的(1,1)点处在屏幕的$(\frac{N-m-1}{2},\frac{N-m-1}{2})$位置，则采用离散傅里叶变换：

$$Y_{p+1,q+1}=\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\omega_n^{jp}\omega_n^{kq}X_{j+1,k+1},0<p,q<N$$

其中$\omega_N=e^{-i\frac{2\pi}{N}}$。X 即为空间复滤波器的矩阵求和范围缩小至算子所在的$m\times m$区域，则：

$$Y_{p+1,q+1}=\sum_{j=N_0}^{N_1}\sum_{k=N_0}^{N_1}\omega_n^{jp}\omega_n^{kq}X_{j-N_0+1,k-N_0+1},0<p,q<N$$

其中$N_0=\frac{N-3-m}{2},N_1=\frac{N-3+m}{2}$。

第二部分 空间复滤波器的振幅和位相分布

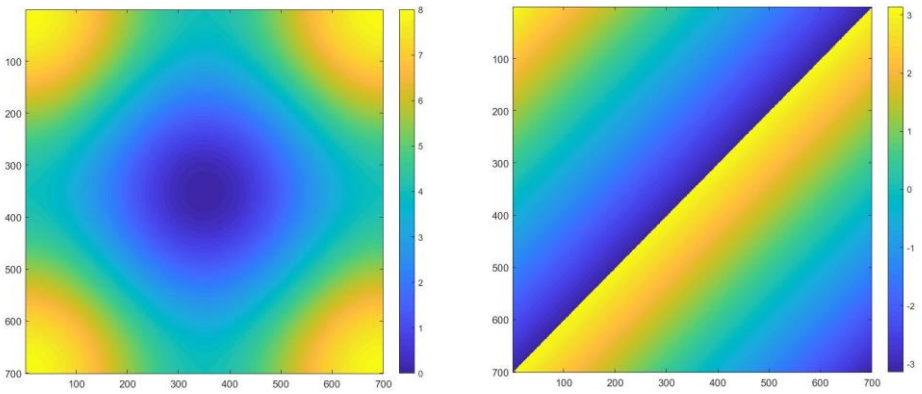
本次程序采用的是 Laplace 算子。写成差分形式为：

$$\nabla^2 f=f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)-4f(x,y)$$

写成算符的形式为：

$$A=\begin{bmatrix}0&1&0\\1&-4&1\\0&1&0\end{bmatrix}$$

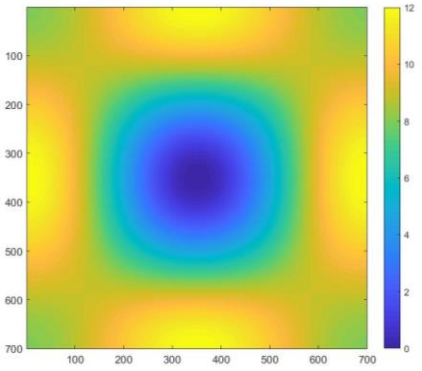
做 FFT,并将 FFT 得到的 H 矩阵设置为与屏幕一样大的 N×N，文件夹中 USTC.jpg 的宽度为 N=700，用到的函数为 fftshift(fft2(A,N,N))，运行得到 H 的振幅和位相分布：

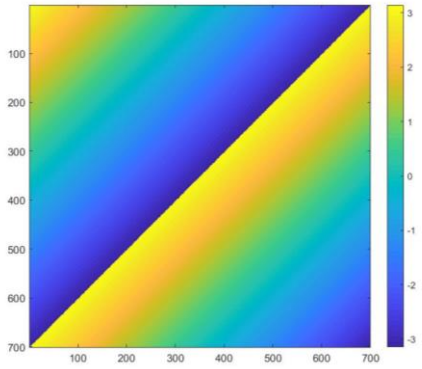


在一些文献中，也用拉普拉斯算子的另一种形式：

$$A=\begin{bmatrix}1 & 1 & 1\\ 1 & -8 & 1\\ 1 & 1 & 1\end{bmatrix}$$

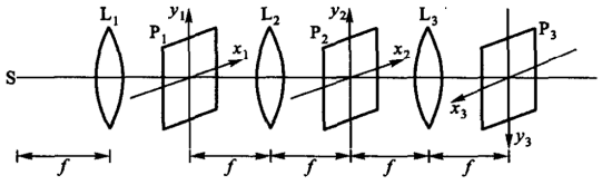
其傅里叶变换后得到的振幅和位相如下：





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$\exp(-\frac{ikr^2}{2z})$。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

```
% 4f system in fourier optics to detect the edge of a p
  icture
clear,clc,clf
Figure=im2double(imread('USTC.jpg'));
Input(:,:)=sqrt(Figure(:,:,3));
nfig=length(Input);
lambda=1;f=1500;stepxy=1;k=2*pi/lambda;nscreen=nfig;
A=[0,1,0;1,-4,1;0,1,0];
H=fftshift(fft2(A,nscreen,nscreen));
x=-(nscreen/2):stepxy:(nscreen/2-stepxy);y=x';
[XX,YY]=meshgrid(x,y);
r=sqrt(XX.^2+YY.^2);
Screen=zeros(nscreen,nscreen);
Screen((nscreen-nfig)/2+1:(nscreen+nfig)/2,(nscreen-
  nfig)/2+1:(nscreen+nfig)/2)=Input;
```

```
U1=RSDiff(f,x,k,Screen);
Phi=exp(-1i*k*r.^2/(2*f));
imshow(abs(U1).^2)
U11=U1.*Phi;
U2=RSDiff(f,x,k,U11);
figure;
imshow(abs(U2).^2);
U22=U2.*H;
figure;
imshow(abs(U22).^2)
U23=RSDiff(f,x,k,U22);
U24=U23.*Phi;
U3=RSDiff(f,x,k,U24);
figure;
imshow(abs(U3).^2);
figure;
imshow(abs(conv2(Input,A,'same')).^2);
```

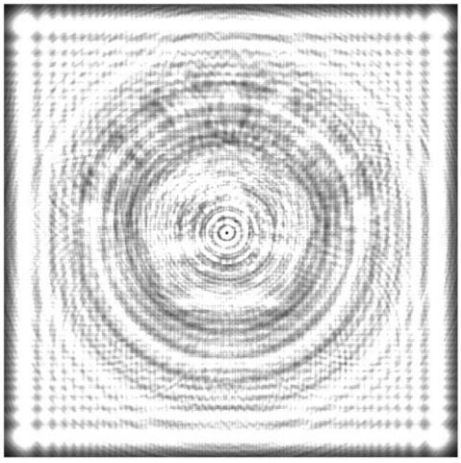
其中 RSDiff.m 函数如下：

```
function Out = RSDiff(z,s,k,object)
%function RSDiff uses the formula in article2006 to cal
culate image
eta=s;x=s;y=s;
n=length(s);
U=[object,zeros(n,n-1);zeros(n-1,n),zeros(n-1,n-1)];
X=[x(1)-s(n+1-(1:(n-1))),x((n:2*n-1)-n+1)-s(1)];
Y=[y(1)-eta(n+1-(1:(n-1))),y((n:2*n-1)-n+1)-eta(1)];
[XX,YY]=meshgrid(X,Y);
r=sqrt(XX.^2+YY.^2+z^2);
G=1/(2*pi)*exp(1i*k*r)./r*z./r.*(1./r-1i*k);
S=ifft2(fft2(U).*fft2(G));
Out=S(n:end,n:end);
end
```

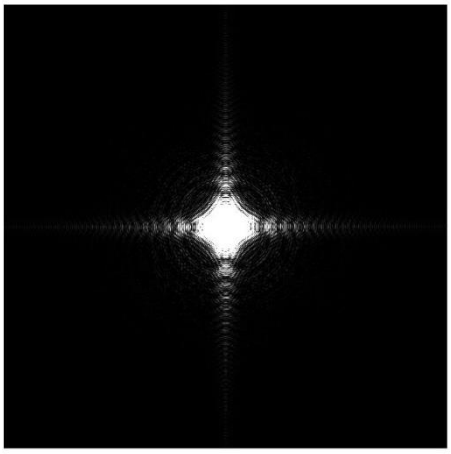
运行结果：

待处理的图像（大小为 700×700 像素）：

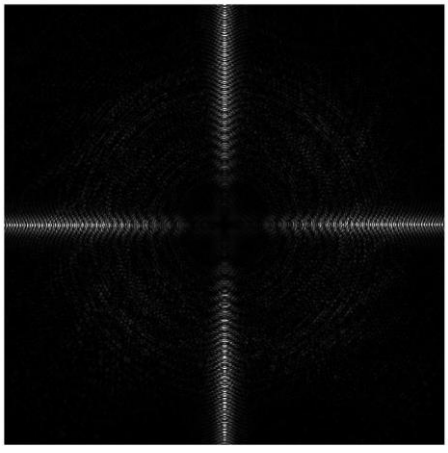
传播至 U1 的图像：



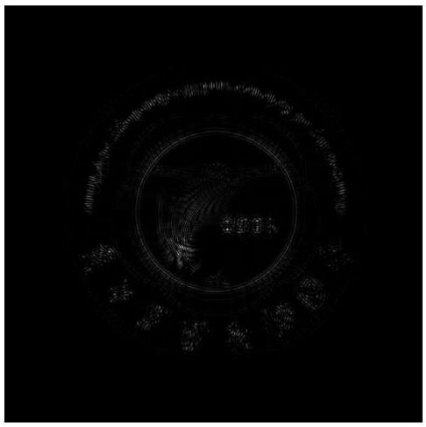
在透镜后焦平面的成像（即上图的频谱）



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：

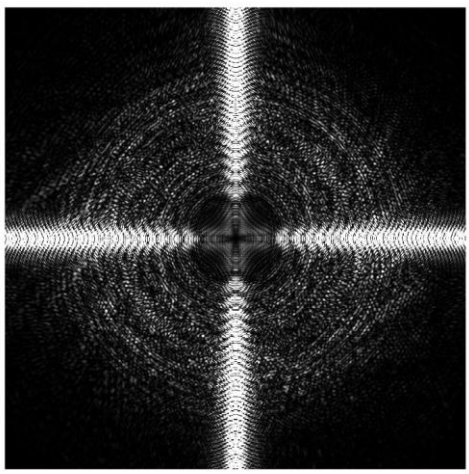


与下图直接用 conv2 函数卷积相比：

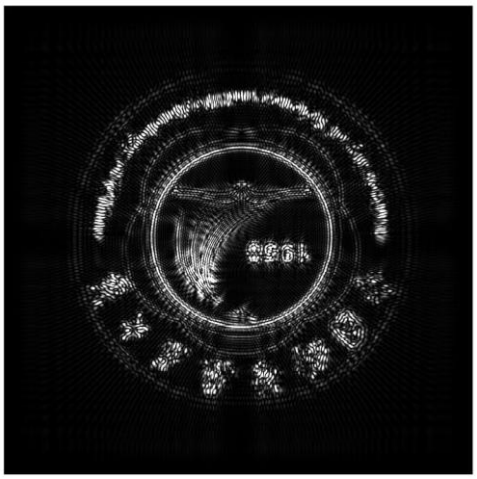
如果改用第二种形式的 Laplace 算子矩阵，即：

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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。