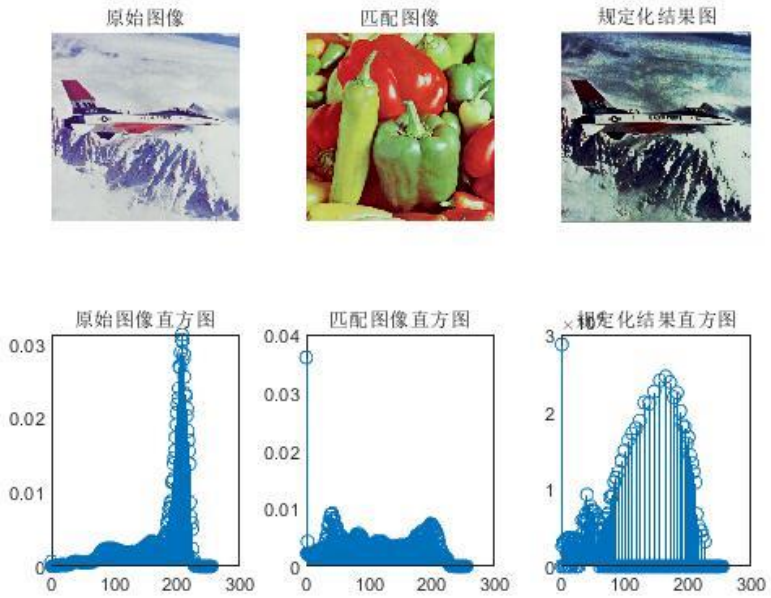
1、 编程实现彩色图像 1 的直方图规定化，并进行结果分析。

1） **结果图：**



2） **结果分析：**

模板图像是刻板化，也是高频化的，通过规定化，原始图像和规定化图像基本上是类似的。

彩色图像需要单独把 RGB 图像的通道提取出来，规定化之后，再把三个通道合起来，形成彩色图像的规定化，规定化的效果相当于图像增强，给图像加滤镜。

3） **源程序：**

```
p = imread('airplane.tiff');
[srcHist,srcX] = imhist(p,256);
srcHist = srcHist/numel(p);
cps = zeros(256,1,'double');
for i=1:1:256
    cps(i) = sum(srcHist(1:i));
end
cps = 255*cps;
cps = uint8(cps);

P1 = imread('peppers.tiff');
[dstHist,dstX] = imhist(P1,256);
dstHist = dstHist/sum(dstHist);
cpd = zeros(256,1,'double');
```

```
for i=1:1:256
    cpd(i) = sum(dstHist(1:i));
end
cpd = cpd*255;
cpd = uint8(cpd);

srcl=zeros(256,1,'uint8');
minv = 256;
for i = 1:1:256
    minv =256;
    for j = 1:1:256
        if minv > abs(cps(i)-cpd(j))
            minv =  abs(cps(i)-cpd(j));
            srcl(i) =j;
        end
    end
end
[width,height] = size( p);
gray1 = p;
for i=1:1:width
    for j = 1:1:height
        gray1(i,j)=srcl(p(i,j)+1);
    end
end

[g1Hist,g1X] = imhist(gray1);
% g1Hist = g1Hist/sum(g1Hist);
g2 = histeq(p,dstHist);
[g2Hist,g2X] = imhist(g2);
% g2Hist = g2Hist/sum(g2Hist);
figure(1),
subplot(2,3,1),imshow(p),title('原始图像');
subplot(2,3,2),imshow(P1),title('匹配图像');
subplot(2,3,3),imshow(gray1),title('规定化结果图');
subplot(2,3,4),stem(srcX,srcHist),title('原始图像直方图');
subplot(2,3,5),stem(dstX,dstHist),title('匹配图像直方图');
subplot(2,3,6),stem(g1X,g1Hist),title('规定化结果图直方图');
```

2、 创建用户交互界面并实现不同方法的空间平滑滤波图像增强和锐化滤波图像增强算法，并进行结果分析。要求交互界面能实现读入图像并可选择进行各种滤波图像增强操作。

1） 空间平滑滤波图像增强和锐化滤波图像增强算法

```
function varargout = JHJM(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @JHJM_OpeningFcn, ...
                   'gui_OutputFcn',  @JHJM_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function JHJM_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

guidata(hObject, handles);
function varargout = JHJM_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function radiobutton1_Callback(hObject, eventdata, handles)
function radiobutton2_Callback(hObject, eventdata, handles)
function radiobutton3_Callback(hObject, eventdata, handles)
function radiobutton4_Callback(hObject, eventdata, handles)

%读入图像
function pushbutton3_Callback(hObject, eventdata, handles)
global s
[filename,pathname,filterindex]=...
uigetfile({'*.*';'*.bmp';'*.tif';'*.png';'*.jpg';'*.jpeg'},'select
picture');
str=[pathname filename];
s=str;
handles.filebig=filterindex;
if filterindex==0
%msgbox('选择图像失败','error');
return
else
im=imread(str);
end
axes(handles.axes1);
```

```
imshow(im);
handles.img=im;
guidata(hObject,handles);

function popupmenu1_Callback(hObject, eventdata, handles)
function popupmenu1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%平滑滤波
function radiobutton5_Callback(hObject, eventdata, handles)
global T
if handles.filebig==0
%msgbox('输入函数图像¡','error');
return;
else
 axes(handles.axes2);
 imshow(handles.img);
T=handles.img;
axes(handles.axes2);
prompt={'请输入模板维度'};
defans={'3'};
p=inputdlg(prompt,'input',1,defans);
p1=str2num(p{1});
h1=fspecial('average',[p1 p1]);
I=imfilter(handles.img,h1);
end
imshow(I);
handles.img=I;
guidata(hObject,handles);

 %锐化sobel滤波
function radiobutton6_Callback(hObject, eventdata, handles)
global T
if handles.filebig==0
%msgbox('输入函数图像¡','error');
return;
else
 axes(handles.axes2);
 imshow(handles.img);
T=handles.img;
axes(handles.axes2);
```

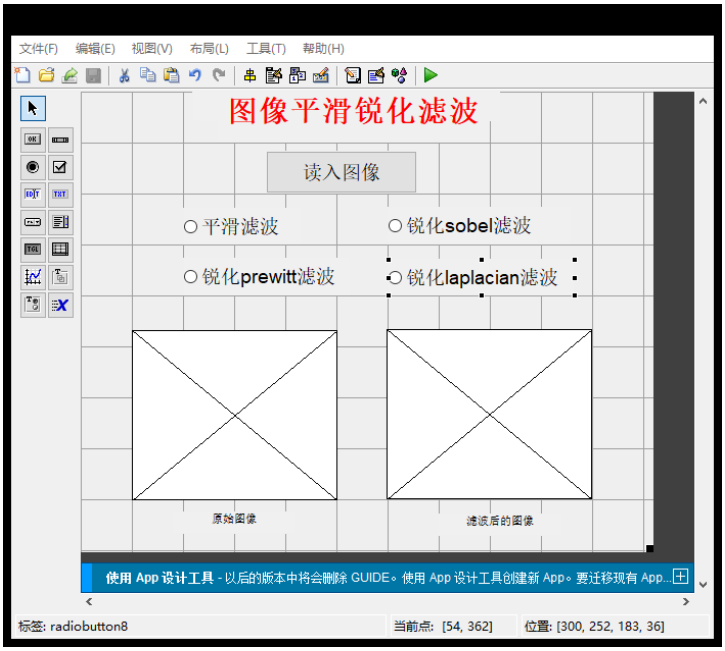
```
h=fspecial('sobel');
g2=imfilter(handles.img,h);
g3=imadd(g2,handles.img);
end
imshow(g3);
handles.img=g3;
guidata(hObject,handles);

%锐化prewitt滤波
function radiobutton7_Callback(hObject, eventdata, handles)
global T
if handles.filebig==0
%msgbox('输入函数图像¡','error');
return;
else
 axes(handles.axes2);
 imshow(handles.img);
T=handles.img;
axes(handles.axes2);
h=fspecial('prewitt');
g2=imfilter(handles.img,h);
g3=imadd(g2,handles.img);
end
imshow(g3);
handles.img=g3;
guidata(hObject,handles);

 %锐化laplacian滤波
function radiobutton8_Callback(hObject, eventdata, handles)
global T
if handles.filebig==0
msgbox('输入函数图像¡','error');
return;
else
 axes(handles.axes2);
 imshow(handles.img);
T=handles.img;
axes(handles.axes2);
h=fspecial('laplacian');
g2=imfilter(handles.img,h);
g3=imadd(g2,handles.img);
imshow(g3);
end
handlse.img=g3;
```

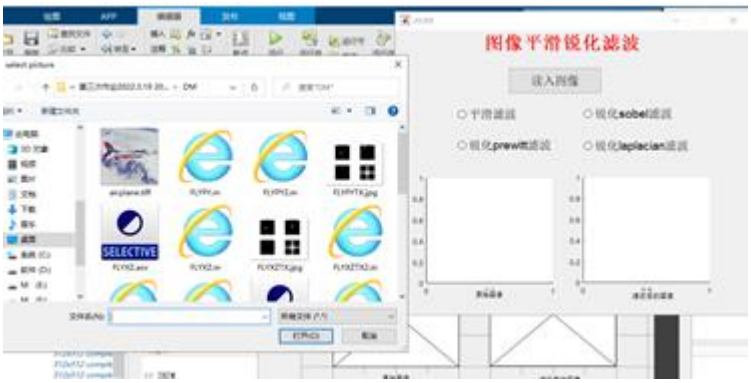
```
guidata(hObject,handles);
```

2） 交互界面截图



3） 结果分析

a） 读入图像



b） 平滑滤波

c） 锐化 sobel 滤波

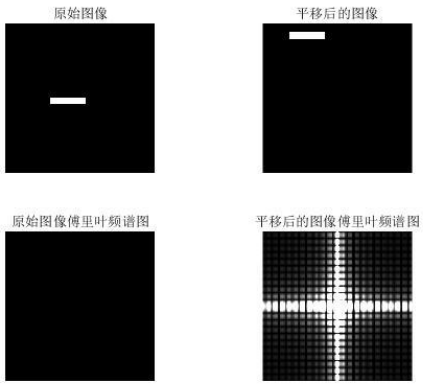
d） 锐化 prewitt 滤波

e） 锐化 laplacian 滤波

3、 给定所示图像，要求编程实现该图像经平移、旋转后的二维傅里叶变换频谱图，并根据频谱图进行结果分析。

1） 图像平移后傅里叶变换频谱图

a） 结果图



b） **结果分析**

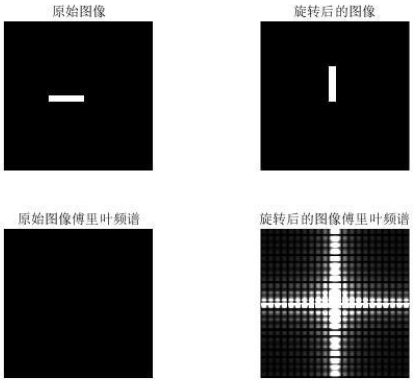
对测试图像进行平移变换后，傅里叶频谱图分布集中在了二维中轴上，有利于对图像进行进一步分析。

c） **源程序**

```
P=imread('测试图像.tiff');
i=double(i)
j=fft2(i);
k=fftshift(j);
l=log(abs(k));
m=fftshift(j);
RR=real(m);
II=imag(m);
A=sqrt(RR.^2+II.^2);
A=(A-min(min(A)))/(max(max(A)))*255;
b=circshift(P,[800 450]);
b=double(b)
c=fft2(b);
e=fftshift(c);
l=log(abs(e));
f=fftshift(c);
WW=real(f);
ZZ=imag(f);
B=sqrt(WW.^2+ZZ.^2);
B=(B-min(min(B)))/(max(max(B)))*255;
subplot(2,2,1);imshow(P);title('原始图像')
subplot(2,2,2);imshow(uint8(b));;title('平移后的图像')
subplot(2,2,3);imshow(A);title('原始图像傅里叶频谱图');
subplot(2,2,4);imshow(B);title('平移后的图像傅里叶频谱图')
```

2） **图像旋转后傅里叶变换频谱图**

a） 结果图



b） 结果分析

对测试图像进行旋转变换后，傅里叶频谱图分布集中在了二维中轴上，有利于对图像进行进一步分析，比如说对图像进行滤波，压缩或者加水印。

c） 源程序

```
P=imread('测试图像.tiff');
i=double(i)
j=fft2(i);
k=fftshift(j);
l=log(abs(k));
m=fftshift(j);
RR=real(m);
II=imag(m);
A=sqrt(RR.^2+II.^2);
A=(A-min(min(A)))/(max(max(A)))*255;
b=imrotate(P, -90);
b=double(b)
c=fft2(b);
e=fftshift(c);
l=log(abs(e));
f=fftshift(c);
WW=real(f);
ZZ=imag(f);
B=sqrt(WW.^2+ZZ.^2);
B=(B-min(min(B)))/(max(max(B)))*255;
subplot(2,2,1);imshow(P);title('原始图像')
subplot(2,2,2);imshow(uint8(b));title('旋转后的图像')
subplot(2,2,3);imshow(A);title('原始图像傅里叶频谱图');
subplot(2,2,4);imshow(B);title('旋转后的图像傅里叶频谱图')
```