可视化作业3

`18300290007 加兴华`

说明：将图片库文件夹放之于路径文件夹下可以保证程序正确运行，smooth_fliter.m，sharpening_fliter.m分别为第一题两种操作的代码。

HW3-1

编程实现图像域基于空间滤波器的（1）平滑操作、（2）锐化算法算法；并把算法应用与图片上，显示与原图的对比差别。

备注：实现的代码不能调用某个算法库里面的函数实现平滑或锐化。

(1)

代码：（matlab）

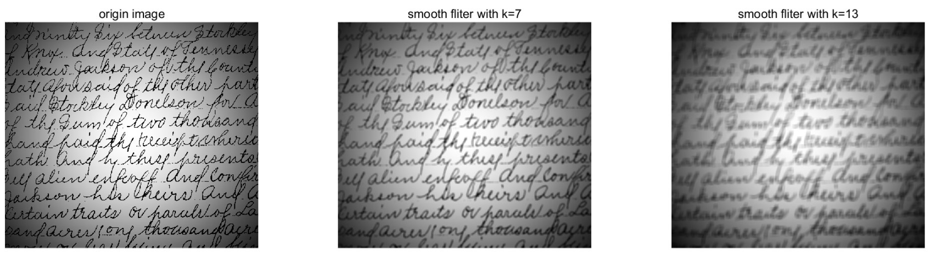
```matlab
clc;clear all;close all
img=imread('图片库//writting.tif');%读取图像信息
subplot(131)
imshow(img)
title('origin image')
b=fill(img,7);
b=smoothfliter(b,7);
subplot(132)
imshow(b)
title('smooth fliter with k=7')
b=fill(img,13);
b=smoothfliter(b,13);
subplot(133)
imshow(b)
title('smooth fliter with k=13')

% 扩展图像来处理边界
function B=fill(A,k) % 采用复制图像边界来扩展
p=(k-1)/2; % 滑窗半径
[m,n,o]=size(A);
% 因为下面生成的扩展矩阵为double类型，为使矩阵乘法合法必须转化
A=im2double(A);
% 扩展矩阵：temp2为行扩展，temp1为列扩展
temp1=blkdiag(ones(1,p+1),eye(n-2),ones(1,p+1));
temp2=blkdiag(ones(p+1,1),eye(m-2),ones(p+1,1));
% 对所有通道操作赋值
for i=1:o
    B(:,:,i)=temp2*A(:,:,i)*temp1;
end
end

% 平滑操作函数
function b=smoothfliter(img,k)
p=(k-1)/2; % 滑窗半径
[m,n,o]=size(img);
% 因为下面生成的滤波器矩阵为double类型，为使矩阵乘法合法必须转化
img=im2double(img);
```

```
% 原尺寸
M=m-2*p;
N=n-2*p;
b=zeros(M,N,o);
%生成平滑滤波器
temp=ones(k)/9;
% 进行卷积核操作
for i=1:M
    for j=1:N
        for t=1:o
        b(i,j,t)=sum(sum(temp.*img(i:i+2*p,j:j+2*p,t)));
        end
    end
end
% 归一化
for i=1:o
    maxb=max(max(b(:,:,i)));
    minb=min(min(b(:,:,i)));
    b(:,:,i)=(b(:,:,i)-minb)./(maxb-minb);
end
end
```

运行结果：



分析：可以看出，平滑操作在图片上的体现为变得模糊，并且随着平滑滤波器滑窗的增大，对图片的模糊效果也会增强。

(2)

代码：（matlab）

```
clc;clear all;close all
img=imread('图片库//moon.tif');%读取图像信息
subplot(231)
imshow(img)
title('origin image')
```

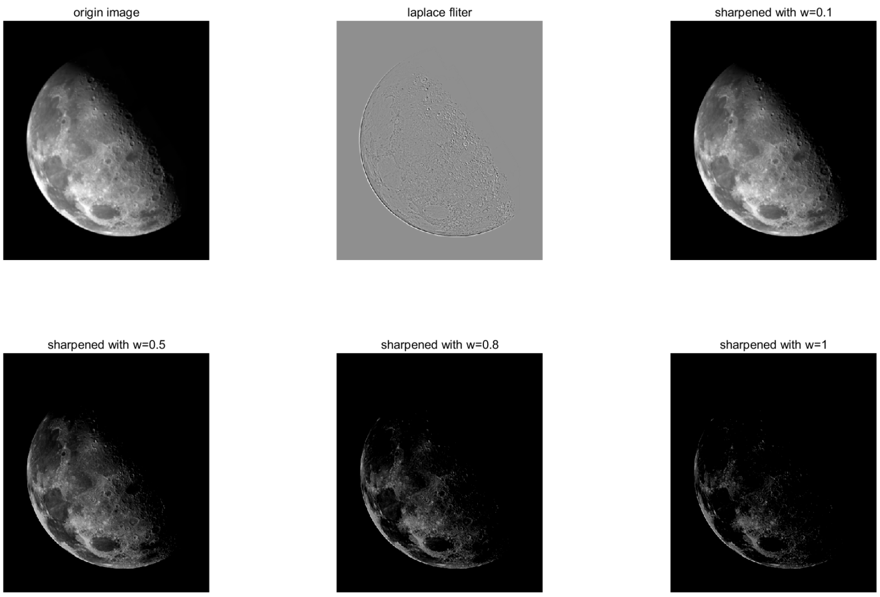
```
b=fill(img);
b=laplacefliter(b);
subplot(232)
imshow(b)
title('laplace fliter')
img=im2double(img);
subplot(233)
imshow(img-0.1*b)
title('sharpened with w=0.1')
subplot(234)
imshow(img-0.5*b)
title('sharpened with w=0.5')
subplot(235)
imshow(img-0.8*b)
title('sharpened with w=0.8')
subplot(236)
imshow(img-1*b)
title('sharpened with w=1')

% 扩展图像来处理边界
function B=fill(A) % 采用复制图像边界来扩展
k=3; % 锐化滑窗为3
p=(k-1)/2; % 滑窗半径
[m,n,o]=size(A);
% 因为下面生成的扩展矩阵为double类型，为使矩阵乘法合法必须转化
A=im2double(A);
% 扩展矩阵：temp2为行扩展，temp1为列扩展
temp1=blkdiag(ones(1,p+1),eye(n-2),ones(1,p+1));
temp2=blkdiag(ones(p+1,1),eye(m-2),ones(p+1,1));
% 对所有通道操作赋值
for i=1:o
    B(:,:,i)=temp2*A(:,:,i)*temp1;
end
end

% 获得拉普拉斯滤波
function b=laplacefliter(img)
k=3; % 锐化滑窗为3
p=(k-1)/2; % 滑窗半径
[m,n,o]=size(img);
% 因为下面生成的滤波器矩阵为double类型，为使矩阵乘法合法必须转化
img=im2double(img);
% 原尺寸
M=m-2*p;
N=n-2*p;
b=zeros(M,N,o);
% 生成图像二阶导数滤波器
temp=zeros(k);
temp(p+1,1:k)=ones(1,k);
temp(1:k,p+1)=ones(k,1);
temp(p+1,p+1)=2-2*k;
% 进行卷积核操作
for i=1:M
    for j=1:N
        for t=1:o
        b(i,j,t)=sum(sum(temp.*img(i:i+2*p,j:j+2*p,t)));
        end
    end
```

```
end
% 归一化
for i=1:o
    maxb=max(max(b(:,:,i)));
    minb=min(min(b(:,:,i)));
    b(:,:,i)=(b(:,:,i)-minb)./(maxb-minb);
end
end
```

运行结果：



分析：可以看出，原图通过减去合适的拉普拉斯滤波会使得轮廓更加鲜明，但是随着拉普拉斯滤波权重的增大，原图却也只剩下轮廓的细节，所以权重并非越大越好，在我使用的图中，我认为w=0.1到0.5之间是最合适的。

HW3-2

证明：

（1）证明冲击窜（impulse train）的傅里叶变换后的频域表达式也是一个冲击窜。

（2）证明实信号f(x)的离散频域变换结果是共轭对称的。

（3）证明二维变量的离散傅里叶变换的卷积定理。

(1)

1.

$S_{\Delta T}(t)=\sum_{-\infty}^{+\infty}c_n e^{\frac{j2\pi}{\Delta T}nt}$ （$\because S_{\Delta T}(t)$是周期为ΔT的函数，$\therefore$ 可以展开成$w=\frac{2\pi}{\Delta T}$的傅里叶级数）

式中$c_n=\frac{1}{\Delta T}\int_{-\frac{\Delta T}{2}}^{\frac{\Delta T}{2}}S_{\Delta T}(t)\,e^{-\frac{j2\pi}{\Delta T}nt}dt=\frac{1}{\Delta T}$

$\therefore F(S_{\Delta T}(t))=\frac{1}{\Delta T}F\left(\sum_{-\infty}^{+\infty}e^{\frac{j2\pi}{\Delta T}nt}\right)=\frac{1}{\Delta T}\sum_{-\infty}^{+\infty}F\left(e^{\frac{j2\pi}{\Delta T}nt}\right)$ （$\because$ 傅里叶变换的线性性）

$=\frac{1}{\Delta T}\sum_{-\infty}^{+\infty}\delta(u-\frac{n}{\Delta T})$ （ $\because F\left(e^{\frac{j2\pi}{\Delta T}nt}\right)=\int_{-\infty}^{+\infty}e^{-j2\pi\left(u-\frac{n}{\Delta T}\right)t}dt=\infty\cdot 1_{(u=\frac{n}{\Delta T})}=\delta(u-\frac{n}{\Delta T})$ ）

(2)

2.

$$F_m=\frac{1}{M}\sum_{n=0}^{M-1}f_n e^{-\frac{j2\pi mn}{M}}\quad,\qquad m=0:M-1$$

$$F_{-m}^{*}=\frac{1}{M}\sum_{n=0}^{M-1}f_n e^{\frac{j2\pi(-m)n}{M}}\quad,\qquad m=0:M-1$$

$\therefore F_{-m}^{*}=F_m$，得证

(3)

3.

二维离散傅里叶变换：F(u,v)=$F(f)=\frac{1}{MN}\sum_0^{M-1}\sum_0^{N-1}f(m,n)\,e^{-\frac{j2\pi mu}{M}}e^{-\frac{j2\pi nv}{N}}$

一方面：

$$F(f*h)=\sum_0^{M-1}\sum_0^{N-1}f(m,n)*h(m,n)e^{-\frac{j2\pi mu}{M}}e^{-\frac{j2\pi nv}{N}}$$

$$=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left\{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\cdot h(m-x,n-y)\right\}e^{-\frac{j2\pi mu}{M}}e^{-\frac{j2\pi nv}{N}}$$

$$=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\cdot h(m-x,n-y)\,e^{-\frac{j2\pi mu}{M}}e^{-\frac{j2\pi nv}{N}}$$

$$=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)e^{-\frac{j2\pi xu}{M}}e^{-\frac{j2\pi yv}{N}}\right]\cdot\left[h(m-x,n-y)\,e^{-\frac{j2\pi(m-x)u}{M}}e^{-\frac{j2\pi(n-y)v}{N}}\right]$$

$$=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)e^{-\frac{j2\pi xu}{M}}e^{-\frac{j2\pi yv}{N}}\right]\cdot\left\{\sum_{m=-x}^{M-1-x}\sum_{n=-y}^{N-1-y}h(m,n)\,e^{-\frac{j2\pi mu}{M}}e^{-\frac{j2\pi nv}{N}}\right\}$$

$$=F(h)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)e^{-\frac{j2\pi xu}{M}}e^{-\frac{j2\pi yv}{N}}\right]\quad(\because \text{时移不变性})=F(h)\cdot F(f)$$

另一方面：

$$F^{-1}(F(h)*F(f))=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)*H(u,v)e^{\frac{j2\pi mu}{M}}e^{\frac{j2\pi nv}{N}}$$

$$=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left\{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}F(x,y)\cdot H(u-x,u-y)\right\}e^{\frac{j2\pi mu}{M}}e^{\frac{j2\pi nv}{N}}$$

$$=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[F(x,y)e^{\frac{j2\pi mx}{M}}e^{\frac{j2\pi ny}{N}}\right]\cdot\left\{\sum_{u=-x}^{M-1-x}\sum_{v=-y}^{N-1-y}H(m,n)\,e^{\frac{j2\pi mu}{M}}e^{\frac{j2\pi nv}{N}}\right\}\qquad(\text{与上述同理可得})$$

$$=MN\cdot f(m,n)\cdot h(m,n)$$