HW3

夏浩 19307130268

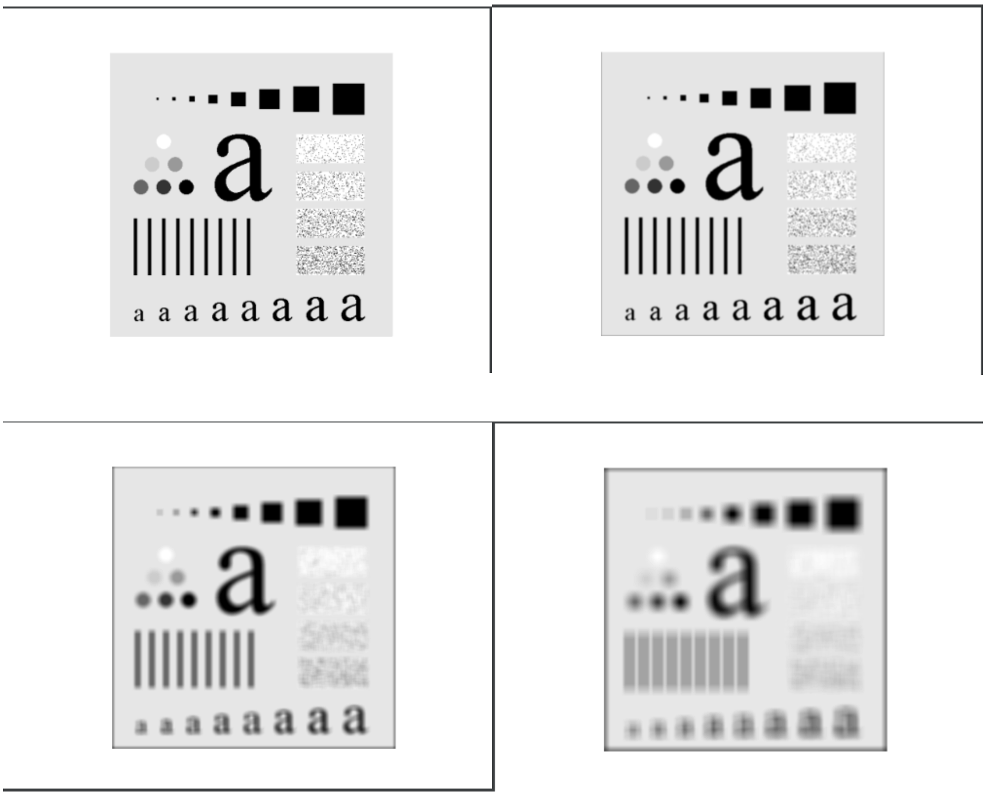
P1.编程实现图像域基于空间滤波器的（1）平滑操作、（2）锐化算法算法；并把算法应用与图片上，显示与原图的对比差别。
备注：实现的代码不能调用某个算法库里面的函数实现平滑或锐化；

(1).

```
import numpy as np
import matplotlib.pylab as plt
import cv2 as cv

def low_filter(im, m=3, n=3):
    h,w = im.shape
    nimg = np.zeros((h, w), np.int16)
    kernel = 1/(m*n) * np.ones((m, n))
    a, b = int((m-1)/2), int((n-1)/2)
    bim = cv.copyMakeBorder(im, a, a, b, b, cv.BORDER_CONSTANT,
value=0)
    for i in range(h):
        for j in range(w):
            nimg[i,j] = np.sum(kernel * bim[i:i+m,j:j+n])
    return  nimg


# load image
img = cv.imread('test1-1.tif', 0)
print(img.shape, img.dtype, type(img))

# show image
plt.imshow(img,cmap = 'gray')
plt.axis('off')
plt.show()

for i in [3,11,21]:
    # process
    nimg = low_filter(img, i, i)
```

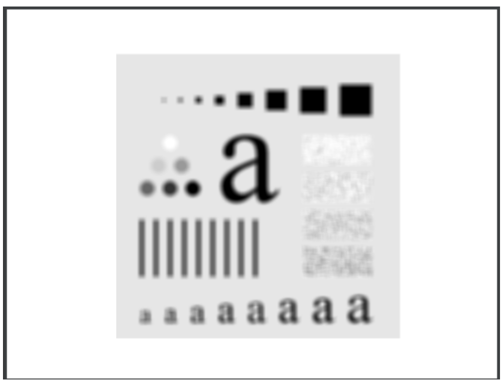
```python
        # show new image
        plt.imshow(nimg,cmap = 'gray')
        plt.axis('off')
        plt.show()
```

下图分别是 原图，$m = n = 3, m = n = 11, m = n = 21$ 的情况，



注意到，直接低通滤波，出现黑色边框，是因为零填充，可通过 [cv.BORDER_REFLECT_101] 作为 参数，实现无黑框的填充。

当$n = 11$,

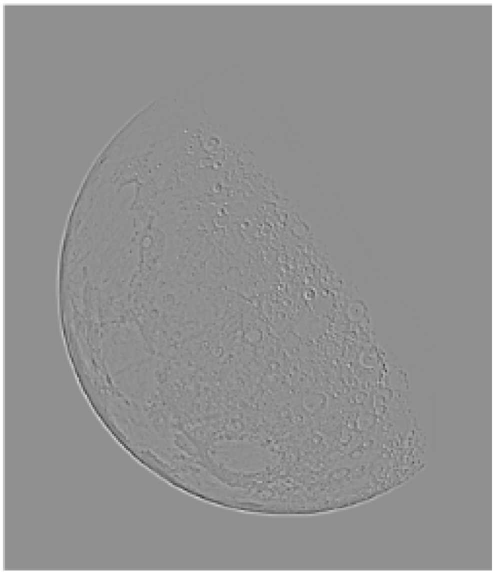


(2).

```python
import numpy as np
```

```python
import matplotlib.pylab as plt
import cv2 as cv

def high_filter(im,kernel):
    h,w = im.shape
    nimg = np.zeros((h, w), np.int16)
    bim = cv.copyMakeBorder(im, 1, 1, 1, 1, cv.BORDER_REPLICATE,
value=0)
    k = 3
    for i in range(h):
        for j in range(w):
            nimg[i,j] = im[i,j] - np.sum(kernel * bim[i:i+k,j:j+k])
    return nimg

# load image
img = cv.imread('test1-2.tif', 0)
print(img.shape, img.dtype, type(img))

# show origin image
plt.imshow(img,cmap = 'gray')
plt.axis('off')
plt.show()

# process
kernel = np.array([[0, 1, 0], [1, -4, 1], [0, 1, 0]])
nimg = high_filter(img,kernel)

# show new image
plt.imshow(nimg, cmap='gray')
plt.axis('off')
plt.show()
```

下图分别为 原图，拉普拉斯图像，锐化图像



[redacted]

P2.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频域/傅里叶变换的卷积定理。

Proof:

(1).

$$
\begin{aligned}
s_{\Delta T}(t) &= \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{\Delta T}t} \\
\text{while,} c_n &= \frac{1}{\Delta T}\int_{-\Delta T/2}^{\Delta T/2} s_{\Delta T}(t) e^{j\frac{2\pi n}{\Delta T}t} dt \\
&= \frac{1}{\Delta T}\int_{-\Delta T/2}^{\Delta T/2} \delta(t) e^{j\frac{2\pi n}{\Delta T}t} \\
&= \frac{1}{\Delta T} \\
s_{\Delta T}(t) &= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi n}{\Delta T}t} \\
\zeta\{e^{j\frac{2\pi n}{\Delta T}t}\} &= \delta(\mu - \frac{n}{\Delta T}) \\
S(\mu) &= \zeta\{s_{\Delta T}(t)\} \\
&= \zeta\{\frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi n}{\Delta T}t}\} \\
&= \frac{1}{\Delta T}\zeta\{\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi n}{\Delta T}t}\} \\
&= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \delta(\mu - \frac{n}{\Delta T})
\end{aligned}
$$

(2).

$$
\begin{aligned}
\tilde{F}(\mu) &= \int_{-\infty}^{\infty} \tilde{f}(t) e^{-j2\pi\mu t} dt \\
&= \sum_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t)\delta(t-n\Delta T) e^{-j2\pi\mu t} dt \\
&= \sum_{-\infty}^{\infty} f(n\Delta T) e^{-j2\pi n\Delta T} \\
\tilde{F}(-\mu) &= \int_{-\infty}^{\infty} \tilde{f}(t) e^{j2\pi\mu t} dt \\
&= \sum_{-\infty}^{\infty} f(n\Delta T) e^{-j2\pi n\Delta T} \\
&= \tilde{F}(\mu)
\end{aligned}
$$

对称

(3).

$$
\begin{aligned}
&(f*h)(x,y) \Leftrightarrow (F\cdot H)(u,v) \\
&\quad \zeta\{f*h(x,y)\} = \zeta(u,v) \\
&\qquad = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)h(x-m,y-n)e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} \\
&\qquad = (\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}) \times (\sum_{x=m}^{M-1}\sum_{y=n}^{N-1} h(x-m,y-n)e^{-j2\pi(\frac{u(x-m)}{M}+\frac{v(y-n)}{N})} dxdy \\
&\qquad = (F\cdot H)(u,v)
\end{aligned}
$$

同理，$(F\cdot H)(u,v) = (f*h)(x,y)$