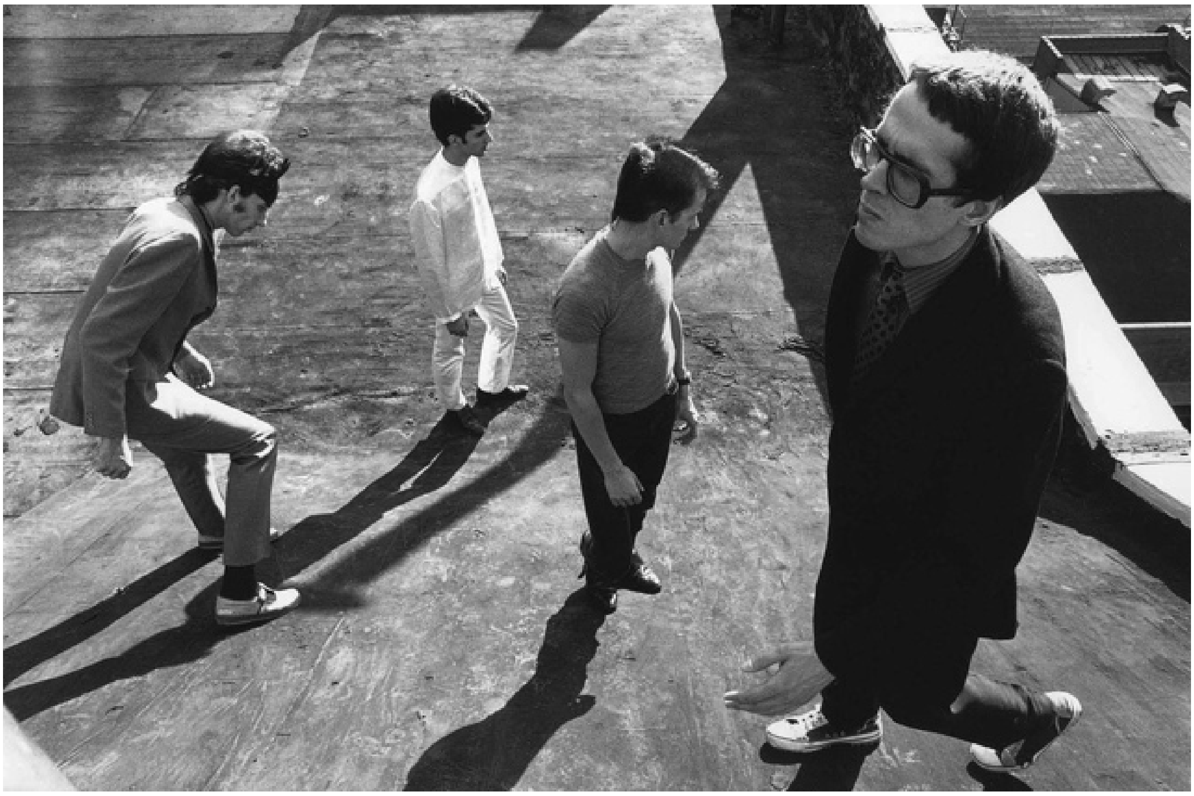
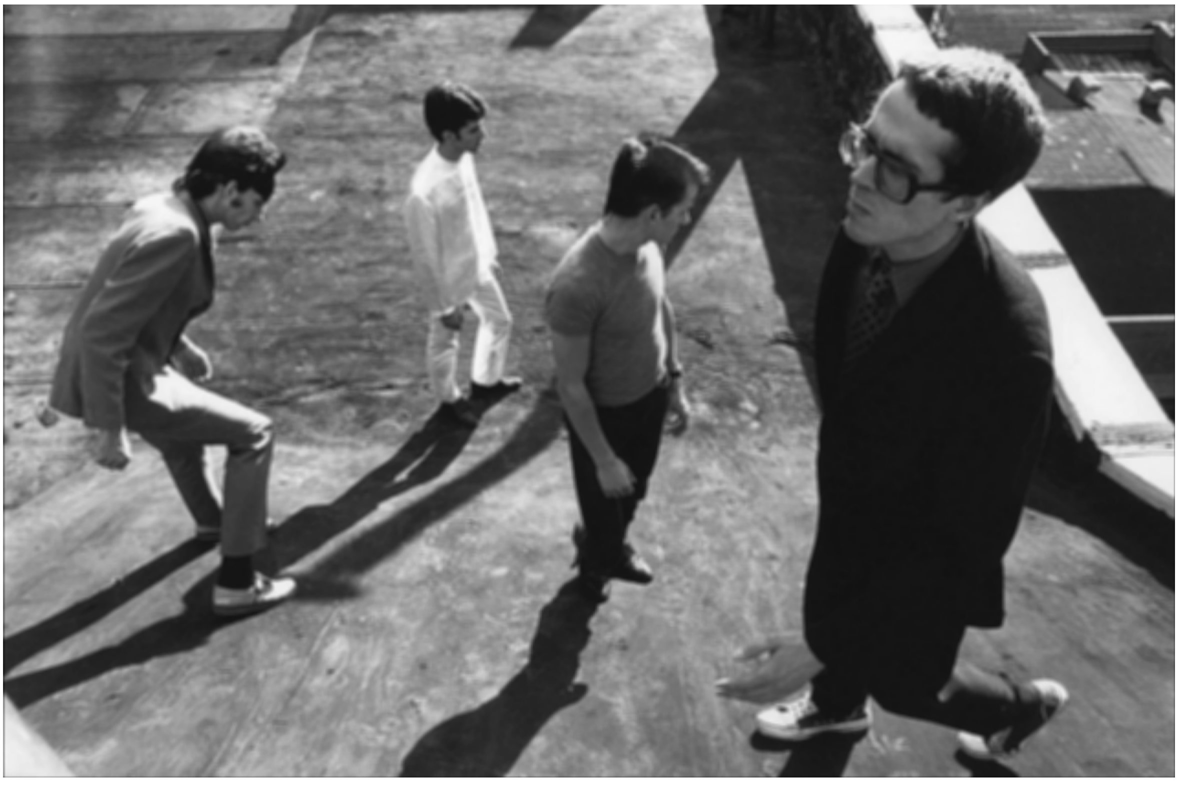
1. Input image

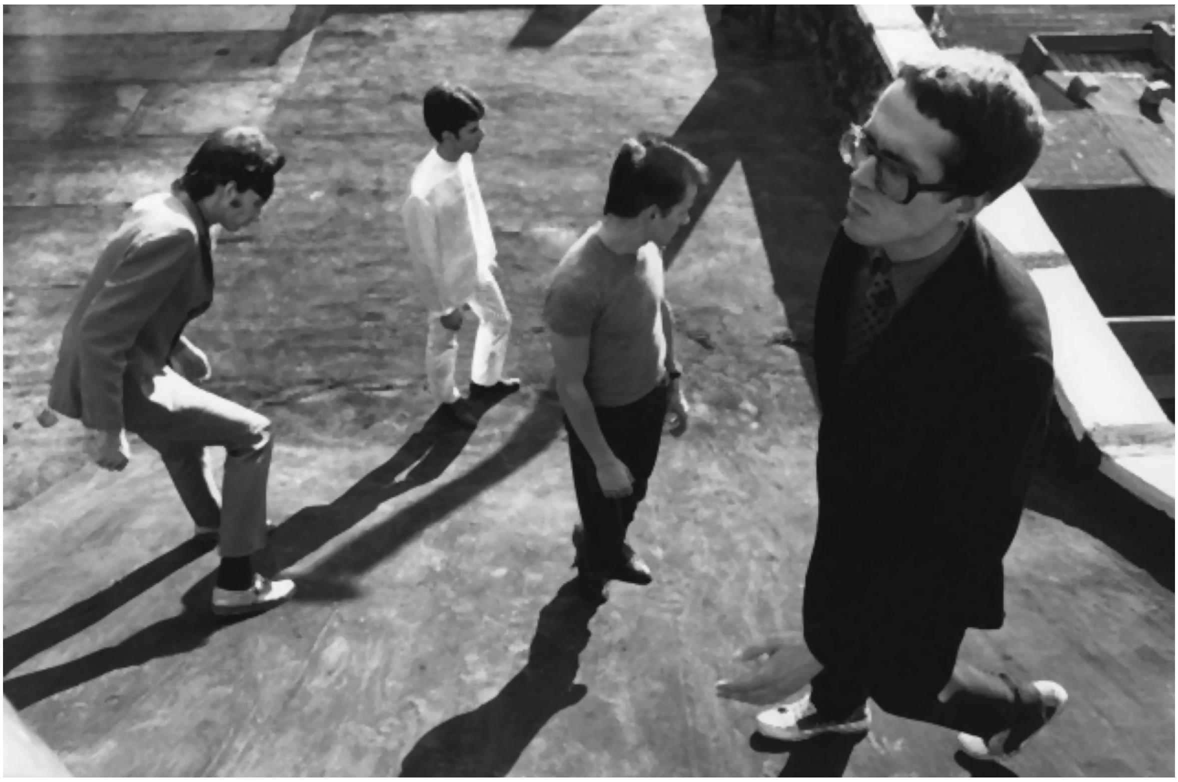


2.

(a) average filter



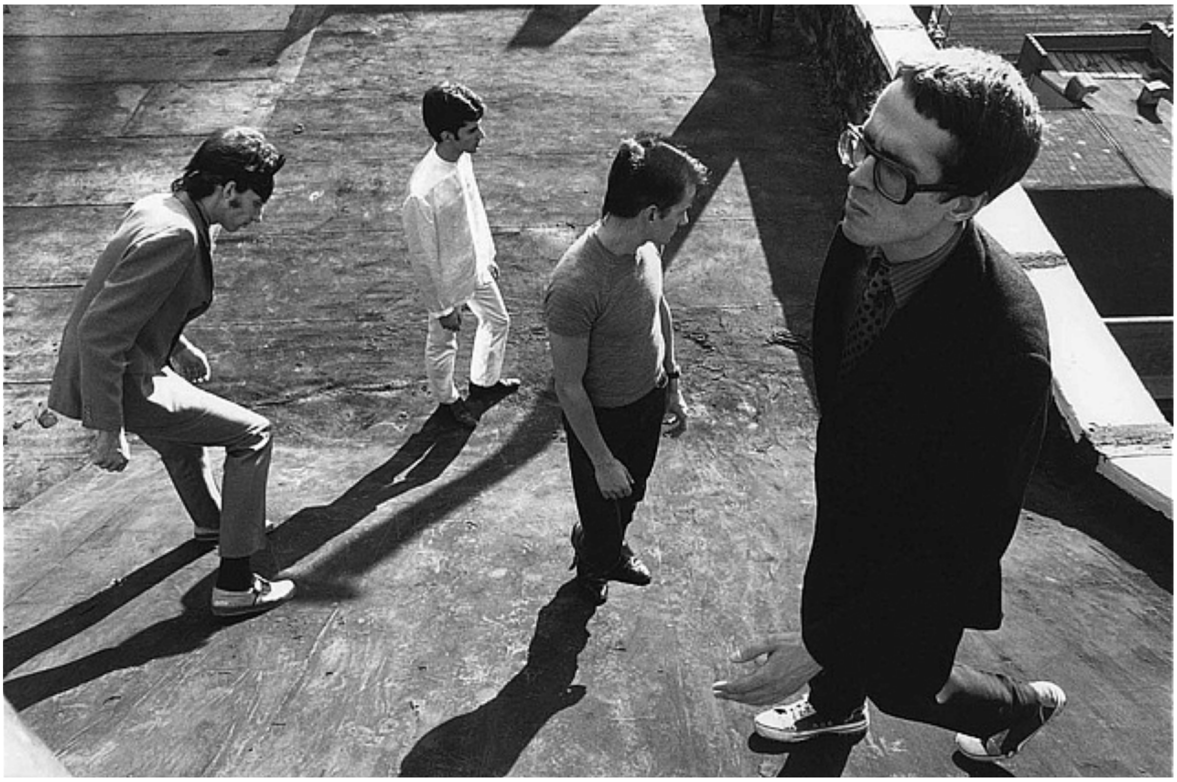
(b) median filter



3. Unsharp masking

k = 0.5

(a)



(b)

source code:

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import cv2

def averageProcess(img, startx, starty, xnum, ynum):
    cnt = 0
    for i in range(ynum):
        for j in range(xnum):
            cnt += int(img[starty + i][startx + j][0])
    res = cnt // 9
    return [res, res, res]


def medianProcess(img, startx, starty, xnum, ynum):
    temp = []
    for i in range(ynum):
        for j in range(xnum):
            temp.append(int(img[starty + i][startx + j][0]))
    temp.sort()
    ret = temp[(len(temp) // 2)]
    return [ret, ret, ret]


# read the image
img = mpimg.imread("input.jpg")

plt.imshow(img)
# disable axis
```

```python
plt.axis('off')
# show the image
plt.show()

y = len(img)
x = len(img[0])

averageFilter = []
medianFilter = []

# get filter data
for i in range(y):

    averageTemp = []
    medianTemp = []

    for j in range(x):
        if i == 0 :
            if j == 0:
                averageTemp.append(averageProcess(img, j, i, 2, 2))
                medianTemp.append(medianProcess(img, j, i, 2, 2))
            elif j == x - 1:
                averageTemp.append(averageProcess(img, j - 1, i, 2, 2))
                medianTemp.append(medianProcess(img, j - 1, i, 2, 2))
            else:
                averageTemp.append(averageProcess(img, j - 1, i, 3, 2))
                medianTemp.append(medianProcess(img, j - 1, i, 3, 2))
        elif i == y - 1:
            if j == 0:
                averageTemp.append(averageProcess(img, j, i - 1, 2, 2))
                medianTemp.append(medianProcess(img, j, i - 1, 2, 2))
            elif j == x - 1:
                averageTemp.append(averageProcess(img, j - 1, i - 1, 2, 2))
                medianTemp.append(medianProcess(img, j - 1, i - 1, 2, 2))
            else:
                averageTemp.append(averageProcess(img, j - 1, i - 1, 3, 2))
                medianTemp.append(medianProcess(img, j - 1, i - 1, 3, 2))
        else :
            if j == 0:
                averageTemp.append(averageProcess(img, j, i - 1, 2, 3))
                medianTemp.append(medianProcess(img, j, i - 1, 2, 3))
            elif j == x - 1:
                averageTemp.append(averageProcess(img, j - 1, i - 1, 2, 3))
                medianTemp.append(medianProcess(img, j - 1, i - 1, 2, 3))
            else:
                averageTemp.append(averageProcess(img, j - 1, i - 1, 3, 3))
                medianTemp.append(medianProcess(img, j - 1, i - 1, 3, 3))

    averageFilter.append(averageTemp)
    medianFilter.append(medianTemp)

# show average filter
plt.imshow(averageFilter)
# disable axis
plt.axis('off')
# show the image
plt.show()

```

```python
# show median filter
plt.imshow(medianFilter)
# disable axis
plt.axis('off')
# show the image
plt.show()

npAverageFilter = np.array(averageFilter)
unsharpByAverage = np.rint(np.rint(img - 0.5 * npAverageFilter) / 0.5)
unsharpByAverage = unsharpByAverage.astype(int)
unsharpByAverage = np.clip(unsharpByAverage, 0, 255)

# show unsharp by average filter image
plt.imshow(unsharpByAverage)
# disable axis
plt.axis('off')
# show the image
plt.show()
# print('done')

npMedianFilter = np.array(medianFilter)
unsharpByMedian = np.rint(np.rint(img - 0.5 * npMedianFilter) / 0.5)
unsharpByMedian = unsharpByMedian.astype(int)
unsharpByMedian = np.clip(unsharpByMedian, 0, 255)

# show unsharp by median filter image
plt.imshow(unsharpByMedian)
# disable axis
plt.axis('off')
# show the image
plt.show()
# print('done')
```

Comment

這次作業花比較多時間在處理形態上的問題，因為 k 是 float，在 show 圖的時候要記得把運算完的 np array 裡面變成 int 且要 clip 進 0 到 255 內。