

《数字图像处理》实验报告

姓名： 汪雨卿 学号： 19120191

实验五

一. 任务 1

自己实现均值滤波，中值滤波，并和程序库自带的版本（Matlab、Pillow、OpenCV 或者其他图像库）进行结果对比。至少在“space.png”“mona.png”上进行测试。

a) 核心代码：

1. 均值滤波

方法一：自己实现的均值滤波

```
def mean_filter(img, size):
    """
    :param img: 输入图像
    :param size: 滤波器大小
    :return: 输出图像
    """
    img_copy = np.copy(img) # 创建读入图像副本
    filt_template = np.ones((size, size)) # 创建空间滤波器模板
    pad_size = int((size - 1) / 2) # 图像需要填充的尺寸
    img_copy = np.pad(img_copy, (pad_size, pad_size), mode="constant", constant_values=0) # 填充图像
    rows, cols = img_copy.shape[:2] # 获取图像大小
    out_img = np.copy(img_copy)

    # 进行空间滤波
    for i in range(pad_size, rows - pad_size):
        for j in range(pad_size, cols - pad_size):
            out_img[i, j] = np.sum(
                filt_template * img_copy[i - pad_size:i + pad_size + 1, j - pad_size:j + pad_size + 1]) / (size ** 2)

    return out_img[pad_size:rows - pad_size, pad_size:cols - pad_size].astype(np.uint8)
```

方法二：调库实现的均值滤波

```
def lib_median(src, k=3):
    img = np.copy(cv.imread(src))
    median_img = ndimage.median_filter(img, k)

    cv.imshow('Before_lib_' + src, img)
    cv.imshow('After_lib_' + src, median_img)
    cv.imwrite('After_lib_' + src, median_img)
    cv.waitKey(0)
    return
```

2. 中值滤波

方法一：自己实现的均值滤波

```
50 def median_filter(src, k=3, padding=None):
51     img = cv.imread(src)
52     img = to_grey(img)
53     img = np.copy(img)
54     cv.imshow('Before_' + src, img)
55     rows, cols = img.shape[:2]
56
57     if not padding:
58         edge = int((k - 1) / 2)
59         if rows - 1 - edge <= edge or cols - 1 - edge <= edge:
60             print("参数k太大,请重新调试")
61             return
62         out_arr = np.zeros((rows, cols), dtype="uint8")
63         for i in range(rows):
64             for j in range(cols):
65                 if i <= edge - 1 or i >= rows - 1 - edge or j <= edge - 1 or j >= cols - edge - 1:
66                     out_arr[i, j] = img[i, j]
67                 else:
68                     out_arr[i, j] = np.median(img[(i - edge):(i + edge + 1), (j - edge):(j + edge + 1)])
69         out_img = out_arr
70     cv.imshow('After_' + src, out_img)
71     cv.imwrite('median_' + src, out_img)
72     cv.waitKey(0)
73     return
```

方法二：调库实现的均值滤波

```
8 def lib_median(src, k=3):
9     img = np.copy(cv.imread(src))
10     median_img = ndimage.median_filter(img, k)
11
12     cv.imshow('Before_lib_' + src, img)
13     cv.imshow('After_lib_' + src, median_img)
14     cv.imwrite('After_lib_' + src, median_img)
15     cv.waitKey(0)
16     return
```

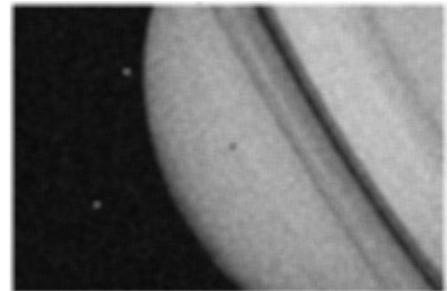
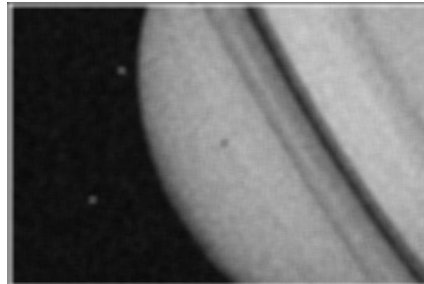
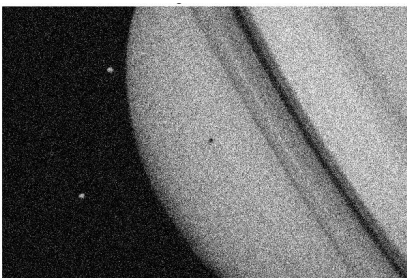
b) 实验结果截图

1. 均值滤波

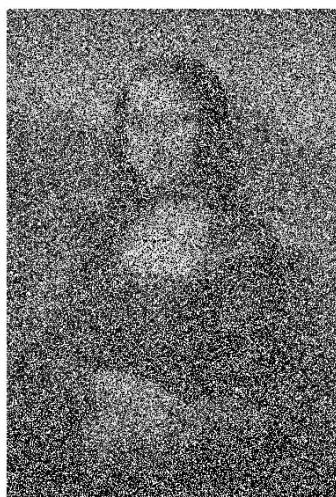
原图：

自己写的均值滤波：

调库做的均值滤波：



原图:



自己写的均值滤波:

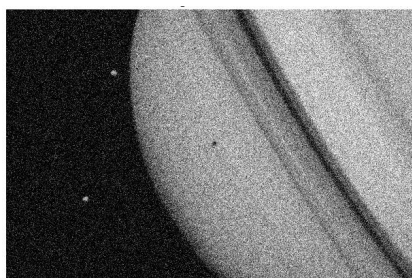


调库做的均值滤波:

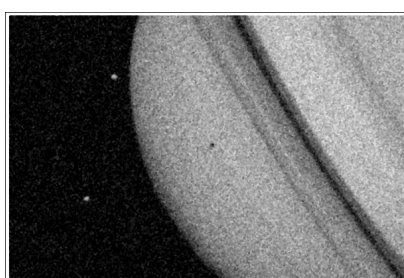


1. 中值滤波

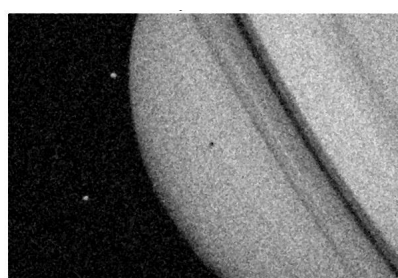
原图:



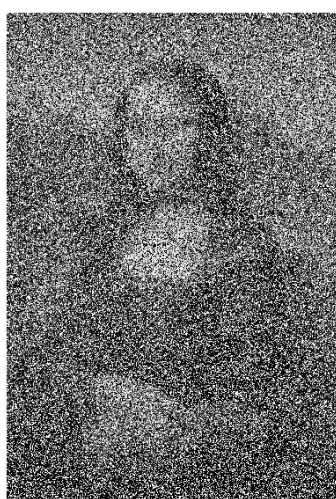
自己写的中值滤波:



调库做的中值滤波:



原图:



自己写的中值滤波:



调库做的中值滤波:



二. 任务 2

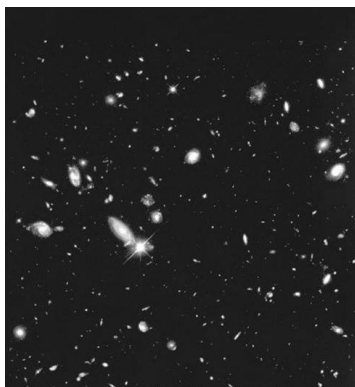
请编写代码，将 star.tif 中较小的行星去除掉（使用平滑滤波器和阈值处理）。

a) 核心代码：

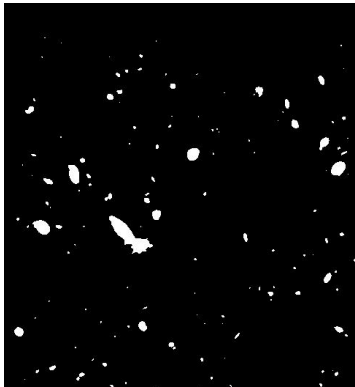
```
def blur_np(img, size = 10):  
    """  
    均值模糊：去随机噪声有很好的去噪效果  
    (1, 15) 是垂直方向模糊，(15, 1) 是水平方向模糊  
    """  
    return cv.blur(img, (size, size))  
  
if __name__ == "__main__":  
    # 将图片转换成RGB  
    star = cv.imread('star.tif', cv.COLOR_BGR2RGB)  
  
    # 调库进行均值滤波  
    filter_star = blur_np(star, 3)  
  
    # 进行阈值调整，大于阈值会取值 maxval 小于阈值取值0 亮越亮 暗越暗  
    ret, thresh_star = cv.threshold(filter_star, 127, 255, cv.THRESH_BINARY)  
    cv.imshow('star', thresh_star)
```

b) 实验结果截图

原图：



修改后



实验小结

本次实验让我学会了通过滤波操作，对图像的某些部门进行过滤，从而实现增强突出所需要的内容，同时弱化无关的图像噪点。最终实现图像的优化。均值滤波可以实现图像的模糊，柔化噪点，但也会造成所需要的信息的模糊。而中值滤波则能够很好的去除校验噪点。在适当的情况下结合使用阈值调整和滤波处理，可以实现图像的优化功能。