

# 数字图像处理

## Pro5-2 Noise Reduction Using a Median Filter

信息与电子工程学院 信息工程

2023 年 4 月 5 日

### 1 实验任务

- (1) 设计 3\*3 中值滤波器;
- (2) 下载图 5.7(a), 添加椒盐噪声  $P_a=P_b=0.2$ ;
- (3) 利用设计的中值滤波器进行滤波, 解释滤波前后和图 5.10(b) 的差异;

### 2 算法设计

中值滤波器的设计公式为  $\hat{f}(x, y) = \text{median}_{(s, t) \in S_{xy}} g(s, t)$ , 即选取一个 3\*3 的矩阵作为邻域后, 选取其中的中值作为目标图像的像素结果。图像的空域卷积处理为了边界不超出范围需要对原图像进行扩展, 因为所用空间滤波器大小为 3\*3, 故扩展行数与列数为原行数列数加 2。matlab 中首先建立扩展后的全零矩阵, 在利用 `padarray` 将原始图像赋值给扩展矩阵。空间滤波可以对图像每一个像素点进行遍历, 在利用 `median` 函数进行选取中值。添加椒盐噪声利用 `imnoise` 函数完成。

### 3 代码实现

---

```
clear; clc; close all
%% 主函数
image= imread('./5_7.TIF');

% 原图像
subplot(1, 3, 1)
imshow(image)
title('original image')

% 加噪图像
image_noise = iamnoise(image, 'salt & pepper', 0.2);
subplot(1, 3, 2)
imshow(image_noise)
```

```

title('image with salt & papper noise')

% 恢复图像
image_filtered = median_filter(image_noise);
subplot(1, 3, 3)
imshow(image_filtered)
title('image after median filter')
%% 3*3 中值滤波器
function image_filter = median_filter(image) % 彩色图中值滤波器

% 获取图像大小
[rows, cols, k] = size(image);

% 创建矩阵, 用于存储结果
image_filter = zeros(rows, cols, 'uint8');

% 对边界像素进行扩展
image_extend = zeros(rows+2, cols+2, 'uint8');
image_extend(:, :) = padarray(image(:, :), [1, 1]);

for i = 1:rows
    for j = 1:cols
        Rect = double(image_extend(i:i+2, j:j+2));
        image_filter(i, j) = uint8(median(Rect, 'all'));
    end
end
end
end

```

---

## 4 实验结果

对于图像 5.7(a), 设置椒盐噪声参数 0.2, 运行上述代码得到结果如下。

原图、加入椒盐噪声以后以及中值滤波后的结果如图所示, 总体上可以完成对椒盐噪声的滤除, 图像也能恢复为原始情况, 只有少量噪点存在。

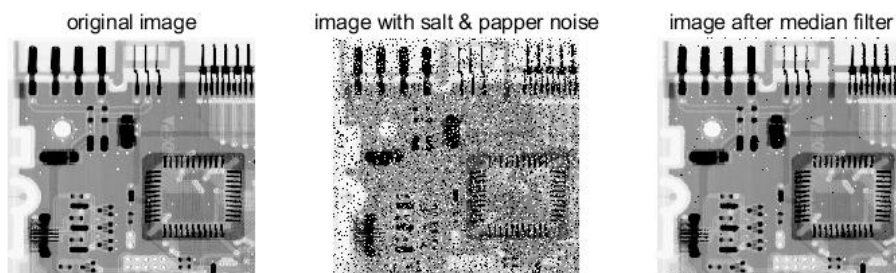


图 1: 中值滤波

图 2 展示了自行设计的中值滤波与标准的中值滤波后的结果, 可以发现几乎没有差别, 设计思路与设计结果都比较理想。

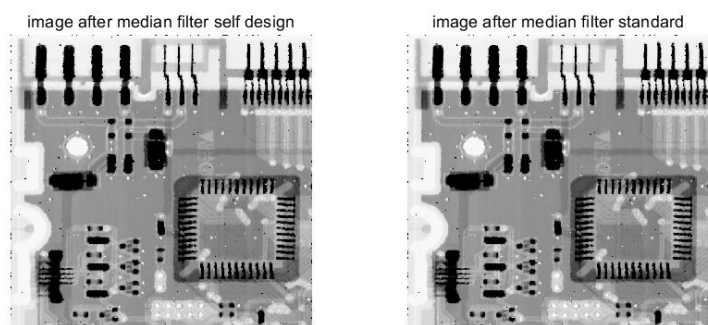


图 2: 与标准中值滤波对比

## 5 总结

本次实验的核心在于空间滤波器的设计。对于  $3 \times 3$  的空间滤波可以用遍历加取中值实现。需要注意的地方有最好能够先定义滤波后的图像矩阵，可以加快代码的运行速度；需要对原图进行范围扩展，以防止超出原图的边界范围。这次实验总体难度不大，对于理解图像的空域处理和卷积操作有很大帮助。