C/C++:中值滤波和均值滤波

2020-10-16 | 图像处理 | 4 |

一、中值滤波

1.原理

针对椒盐噪声设计。因为椒盐噪声的幅值和原图像差异很大，所以在模板中一半分布在最大或最小值附近，而原图像的灰度分布相对接近，通过取模板中间值，可以滤掉椒盐噪声。

2.文件结构

```
main
|
|-read orig img
|
|-convert gray img
|
|-add impulse noise
|
|-middle filt
|
|-result output
```

3.代码实现

```
//中值滤波
cv::Mat middle_filter(cv::Mat &input_img, int size = 3)
{
        cv::Mat output_img = input_img.clone();
        cv::Mat filter = cv::Mat::zeros(1, size*size, CV_8UC1);
        cv::Mat sorted_filter = cv::Mat::zeros(1, size*size, CV_8UC1);
        int M = output_img.rows;
        int N = output_img.cols;
        int offset = (size-1)/2;
        int cnt1, cnt2, cnt3;

        for (cnt1 = offset; cnt1 < M-offset; ++cnt1)
        {
                for (cnt2 = offset; cnt2 < N-offset; ++cnt2)
                {
            //将模板所在位置的图像灰度取出
                        for (cnt3 = 0; cnt3 < size*size; ++cnt3)
                        {
                                filter.at<uchar>(0, cnt3)
                    = input_img.at<uchar>
                        ( cnt1+(cnt3/size-1), cnt2+(cnt3%size-1) );//位置换算
                        }

```

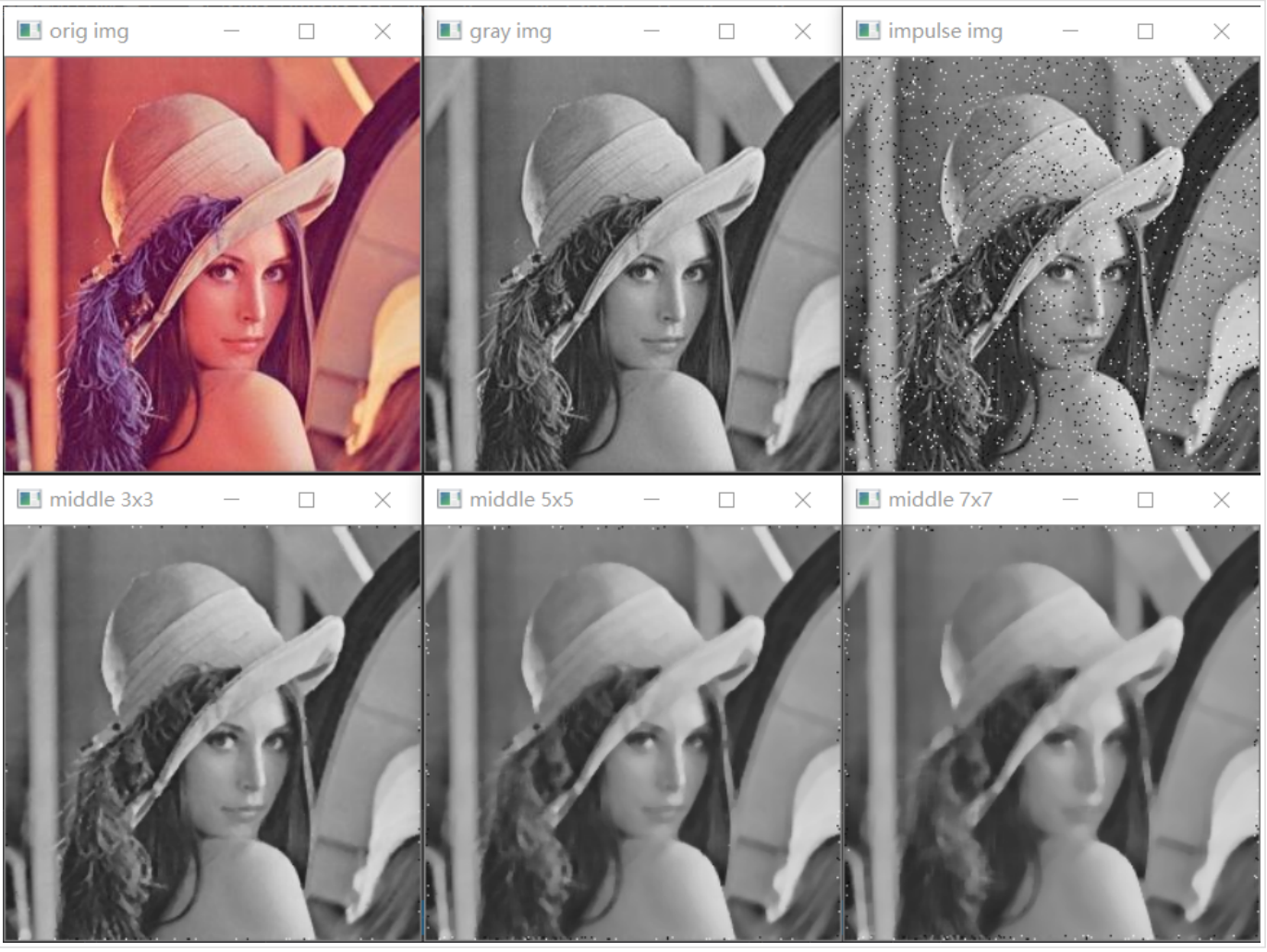
```
            //选择排序
                        sorted_filter = seek_sort(filter);

            //模板中心图像灰度取模板中间值
                        output_img.at<uchar>(cnt1, cnt2)
                = sorted_filter.at<uchar>(0, (size*size-1)/2);
                }
        }

        return output_img;
}
```

4.实验结果

实验结果如下，可以看到椒盐噪声被很好的滤除了（图像边沿保留了原值）。但同时图像也变得模糊，随着模板的增大这一现象更加明显。



二、均值滤波

1.原理

针对高斯噪声设计。滤波器取均值。由于高斯噪声满足高斯分布，所以噪声在每个灰度级上都会存在，但是平均值是一个定值，因此采用平均值滤波可以有效滤除高斯噪声。

2.文件结构

```
main
|
|-read orig img
|
```

```
|-convert gray img
|
|-add gauss noise
|
|-average filt
|
|-result output
```

3.代码实现

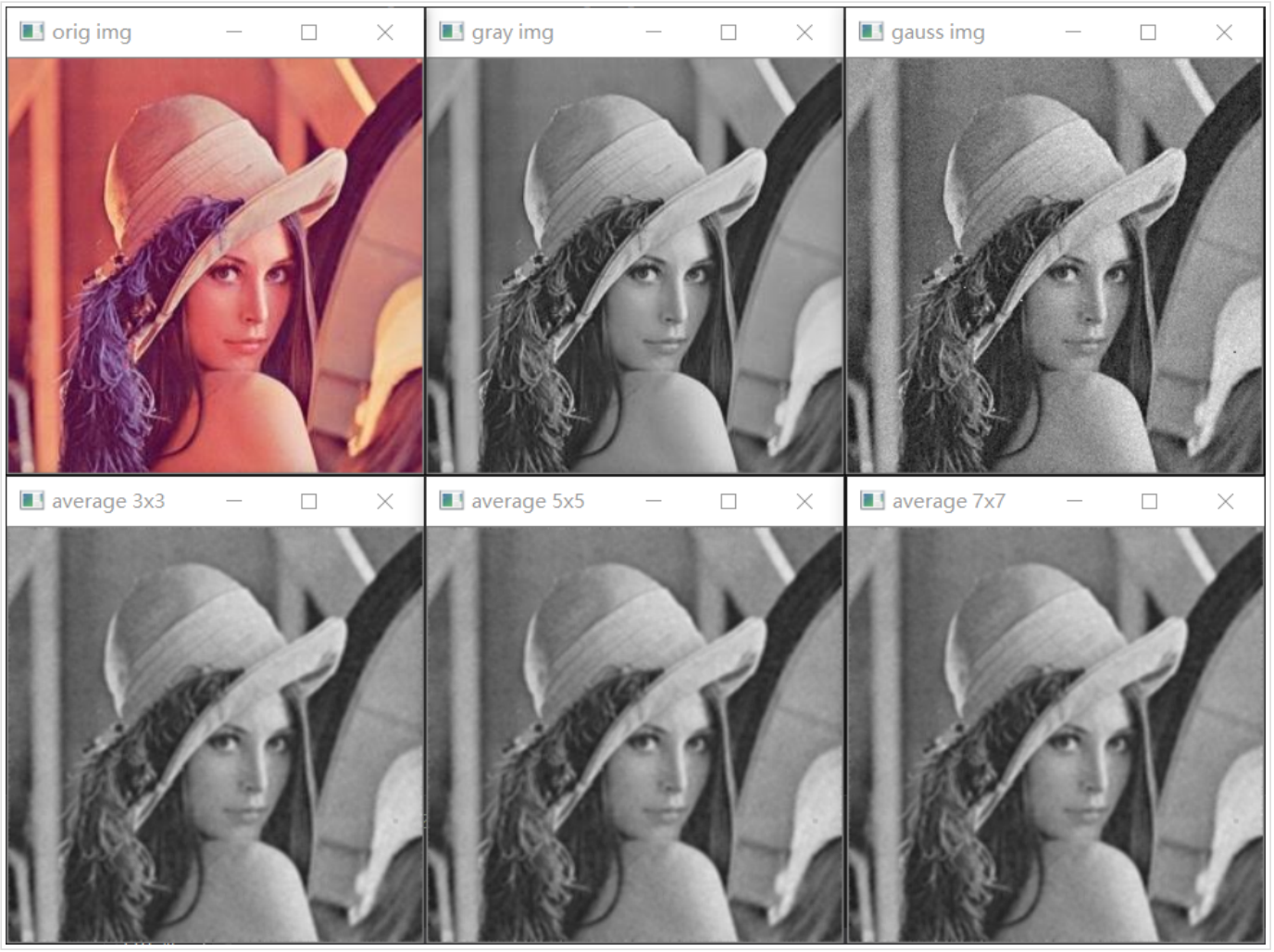
```
//均值滤波
cv::Mat avrg_filter(cv::Mat &input_img, int size = 3)
{
        cv::Mat output_img = input_img.clone();
        cv::Mat filter = cv::Mat::zeros(1, size*size, CV_8UC1);
        cv::Mat sorted_filter = cv::Mat::zeros(1, size*size, CV_8UC1);
        int M = output_img.rows;
        int N = output_img.cols;
        int offset = (size - 1) / 2;
        int cnt1, cnt2, cnt3;
        double avg;

        for (cnt1 = offset; cnt1 < M - offset; ++cnt1)
        {
                for (cnt2 = offset; cnt2 < N - offset; ++cnt2)
                {
                        avg = 0;
            //将模板所在位置的图像灰度取出
                        for (cnt3 = 0; cnt3 < size*size; ++cnt3)
                        {
                                avg += input_img.at<uchar>
                    (cnt1 + (cnt3 / size - offset),
                     cnt2 + (cnt3%size - offset));
                        }
                        avg /= size * size;
                        output_img.at<uchar>(cnt1, cnt2) = (uchar)avg;
                }
        }

        return output_img;
}
```

4.实验结果

实验结果如下，也存在随着模板的增大图像变模糊的现象。

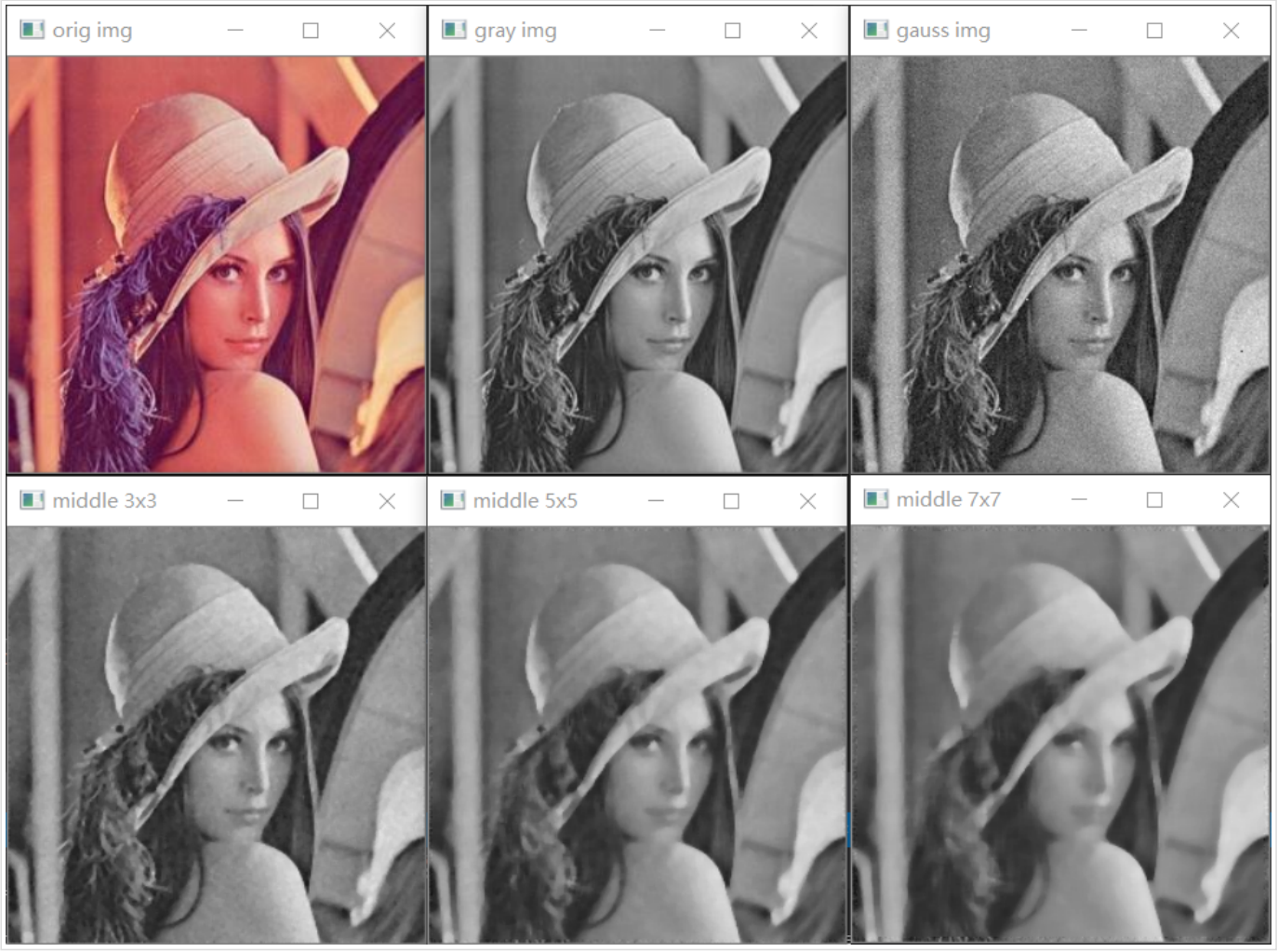


三、滤波效果对比

1.中值滤波对高斯噪声

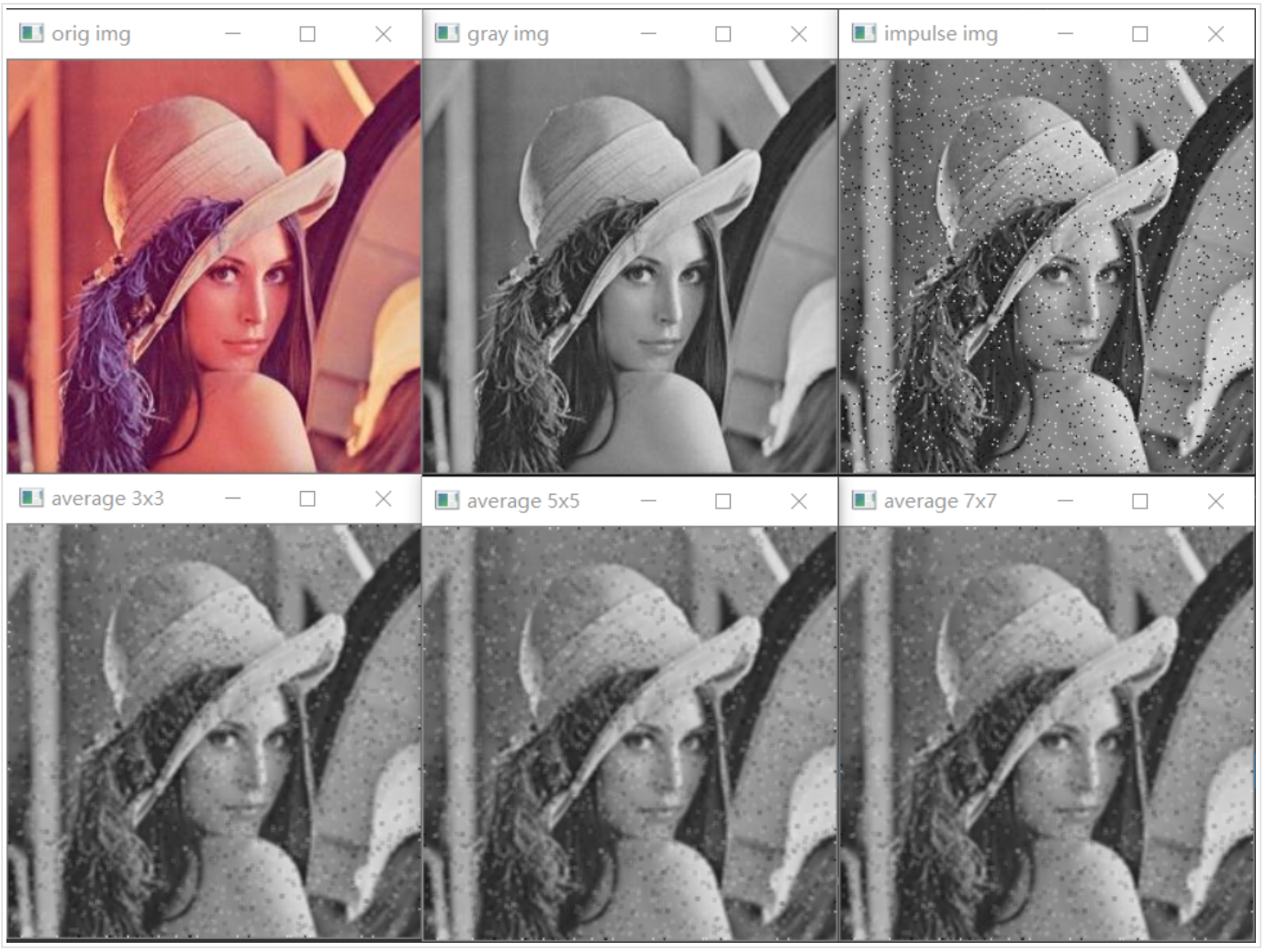
高斯分布均值：0

高斯分布方差：10



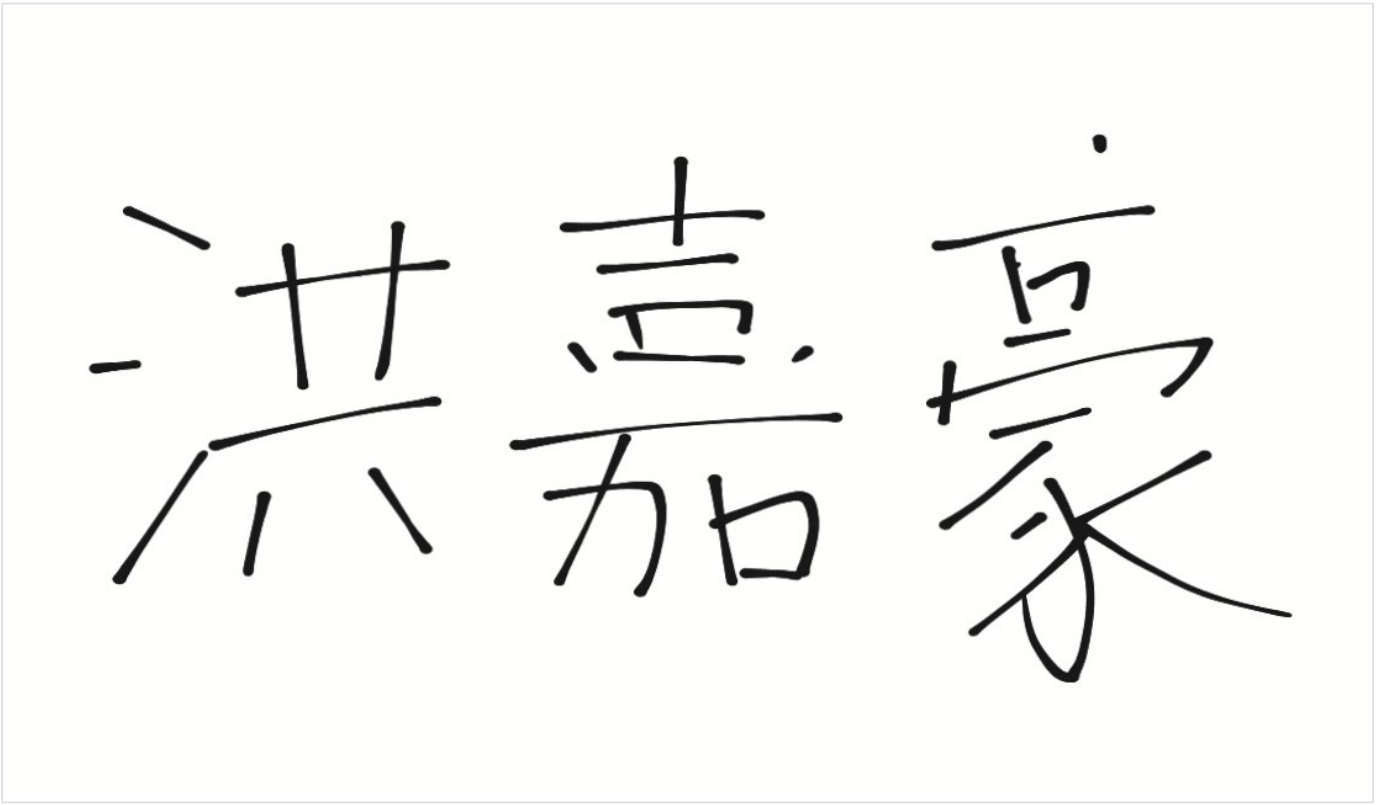
2.均值滤波对椒盐噪声

椒盐噪声信噪比：0.95



小结

由上实验结果可知，中值滤波和均值滤波都有自己的特殊作用，都不是两全其美的滤波方法。



图像处理 # C/C++

Opencv:在VisualStudio里配置Opencv