# Digital Image Processing Course Project Image Restoration Based on Ranked-order Based Adapative Median Filter

Zexi Huang*
Yingcai Honors College

May 21, 2017

**Abstract**

Image restoration is one of the most fundamental topics in digital image processing. Here, we implement a classic algorithm for image restoration, Ranked-order based Adaptive Median Filter (RAMF), which exhibits excellent characteristics in removing positive and negative impulses (salt-and-pepper noises) while simutaneously preserving sharpness. Comparative results with traditional median filter highlight its superiority.

## 1 Introduction

In the field of image restoration, images are often corrupted by salt-and-pepper noises, that is, the positive and negative impulses which are often consequences of decoding errors and noisy channels. An example of corrupted image is shown in Figure 1. An effective noise reduction method for this type of noise is a median filter [1]. However, traditional median filters falter when the probablity of impulse noise occurance becomes high.

In this report, we implement a classic algorithm of adaptive median filter, the Ranked-order based Adaptive Median Filter (RAMF) [2], which exhibits superior results than traditional median filters.

The remainder of this report is organized as follows: In the following section, we illustrate the algorithm and highlight its features. Section 3 presents the numeric results for a specfic benchmark image, with different levels of noises. Finally, we give a short discussion and conclude in Section 4.

## 2 Algorithm

### 2.1 Traditional Median Filter

Prior to the introduction of the RAMF, we first have a brief recollection of the tranditioanl traditional median filter. This filter replaces the value of a pixel by the median of the

*Correspondence should be addressed to Z. Huang. E-mail: Eitima@163.com. Student No. 2014030302014

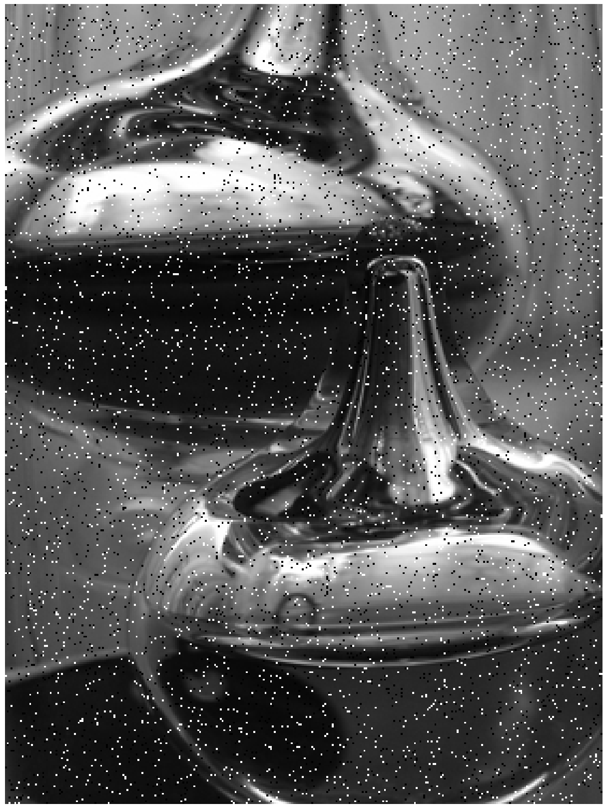

**Figure 1:** An image corrupted by salt-and-pepper noise.

intensity levels in the neighborhood of that pixel:

$$\widehat{f}(x,y) = \operatorname*{median}_{(s,t)\in S_{xy}} f(s,t) \tag{1}$$

where $\widehat{f}(x,y)$ denotes the result of processing, $f(x,y)$ denotes the input corrupted image, and $S_{xy}$ denotes a neighborhood around the pixel $(x,y)$, the size and shape of which can be varied. Here, we only consider the case when $S_{xy}$ is chosen as a rectangular window area.

For salt-and-pepper noises, since the impulses are hardly the median in a neighborhood (they are either the minimum or the maximum, instead), they can always be filtered out. In addition, median filters also enjoy less blurring than linear smoothing filters of similar size.

## 2.2 Ranked-order based Adaptive Median Filter

The traditional median filters discussed in previous subsection performs well if the spatial density of the impulse noises is small (smaller than 0.2, as a rule of thumb). The RAMF introduced in this subsection, however, yields excellent results after this limit and can also preserve details instead of replacing each pixel with the neighborhood median.

Denote: $z_{min}$ as the minimum intensity value in $S_{xy}$, $z_{max}$ as the maximal intensity value in $S_{xy}$, $z_{med}$ as the median of intensity values in $S_{xy}$, $z_{xy}$ as the intensity value at $(x,y)$, $r_{max}$ as the maximal allowed radius of $S_{xy}$.

The algorithm than can be described in Algorithm 1.

**Algorithm 1** Ranked-order based Adaptive Median Filtering

```
for each pixel (x, y) do
    r ← 1, loopFlag ← True
    while loopFlag == True do
        S_xy ← the r-rectangular neighborhood of (x, y)
        if z_med > z_min && z_med < z_max then
            if z_xy > z_min && z_xy < z_max then
                f̂(x, y) ← z_xy, loopFlag ← False
            else
                f̂(x, y) ← z_med, loopFlag ← False
            end if
        else
            if r < r_max then
                r ← r + 1
            else
                f̂(x, y) ← z_med, loopFlag ← False
            end if
        end if
    end while
end for
```

The rationale of this algorithm is to replace impulse noises with median values while keeping other pixels as their original values. The first condition, $z_{med} > z_{min}$ && $z_{med} < z_{max}$, tests whether the median value is an impulse. If it's true, then the median value can't be an impulse and thus can be chosen as the right median value. Otherwise, we should entend the neighborhood area until the median value is not an impulse. The second condition, $z_{xy} > z_{min}$ && $z_{xy} < z_{max}$, determines whether the pixel in the center is an impulse. If it holds, then the center pixel is not an impulse. Therefore, we can directly output this pixel without tuning. Otherwise, it must be an impulse and should be replaced by the median value.

The RAMF approach is thus free from parameter tunning (the radius) and also preserve details even in each neighborhood.

# 3 Numerical Experiments

## 3.1 Parameter Setting and Enviroment

The shape of the neighborhood $S_{xy}$ is chosen to be rectangular. The maximal radius $r_{max}$ is chosen as $r_{max} = 5$, which is considered large enough. The image boundaries are padded with mirror reflections of themselves.

The simulation environment for the whole process is MATLAB 9.0.0.341360 (R2016a), and all sources codes used in this simulation and listed in the appendix are of that syntax. This includes the implementation of RAMF *RAMF.m* and several other auxiliary functions.

### 3.2 Comparative Results

We compare the results of RAMF and traditional median filters under different levels of noises. The original image is one of the widely-used benchmark image, *lena*. The spatial density $p$ of the impulse noises added to this image ranges from 0.1 to 0.5. Three traditional median filters are used for comparation, with their radius $r = 1, 2, 3$ (that is, $3 \times 3$, $5 \times 5$ and $7 \times 7$ masks). The results are generated using *ComparativeAnalysis.m* with respective spatial densities, and are shown in Figure 2, Figure 3, Figure 4, Figure 5 and Figure 6, respectively.

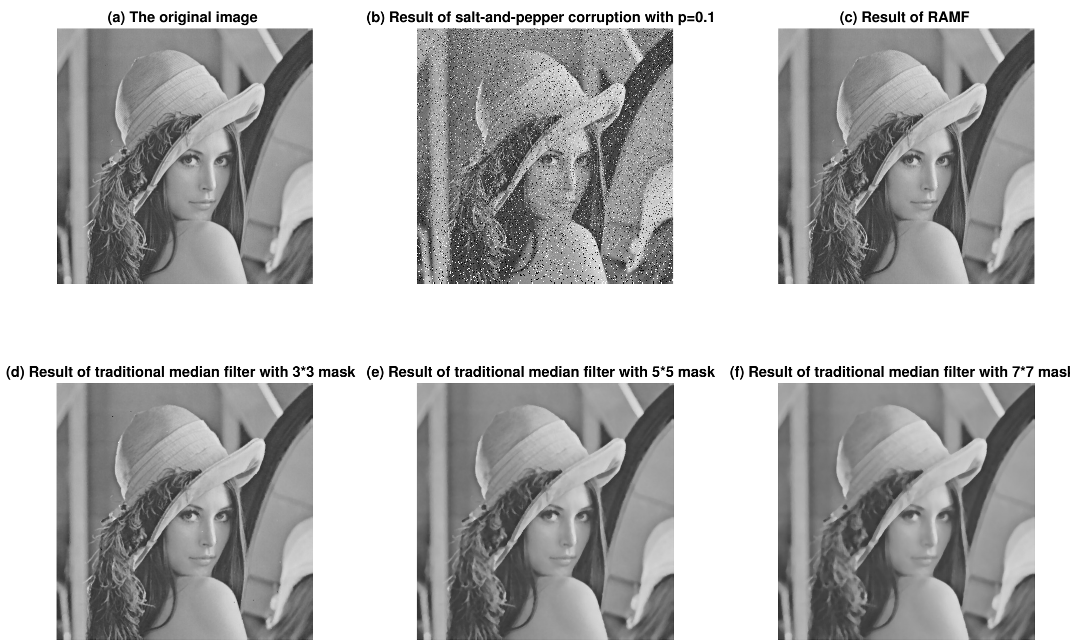


**Figure 2:** Comparison between RAMF and traditional median filters. Spatial density $p = 0.1$. (a) Original image. (b) Result of salt-and-pepper corruption. (c) Result of RAMF. (d) Result of traditional median filter with $3 \times 3$ mask. (e) Result of traditional median filter with $5 \times 5$ mask. (f) Result of traditional median filter with $7 \times 7$ mask.

As shown in Figure 2, with relatively low spatial density of impulses, all filters can eliminate the noise. However, since traditional median filters always replace each pixel with the median value in its neighbourhood, details of the image are lost. This can be seen from (d) to (f). As the mask becomes larger, the image is blurred to an more enormous extent. Since Figure 3 the traditional median filter with $3 \times 3$ mask can't fulfill its duty to eliminate all the impulses, as mentioned before. Since Figure 5 the median filter with $3 \times 5$ mask begin to fail. In addition, although the traditional filter with $7 \times 7$ mask seem to be able to eliminate the impulses even for $p = 0.5$, we can observe that as $p$ increases, the result becomes more and more blurred, and in Figure 6 (f) we can even tell that the image is comprised of blocks of identical pixels.

Compared to the traditional filters, we see that the results from RAMF are excellent

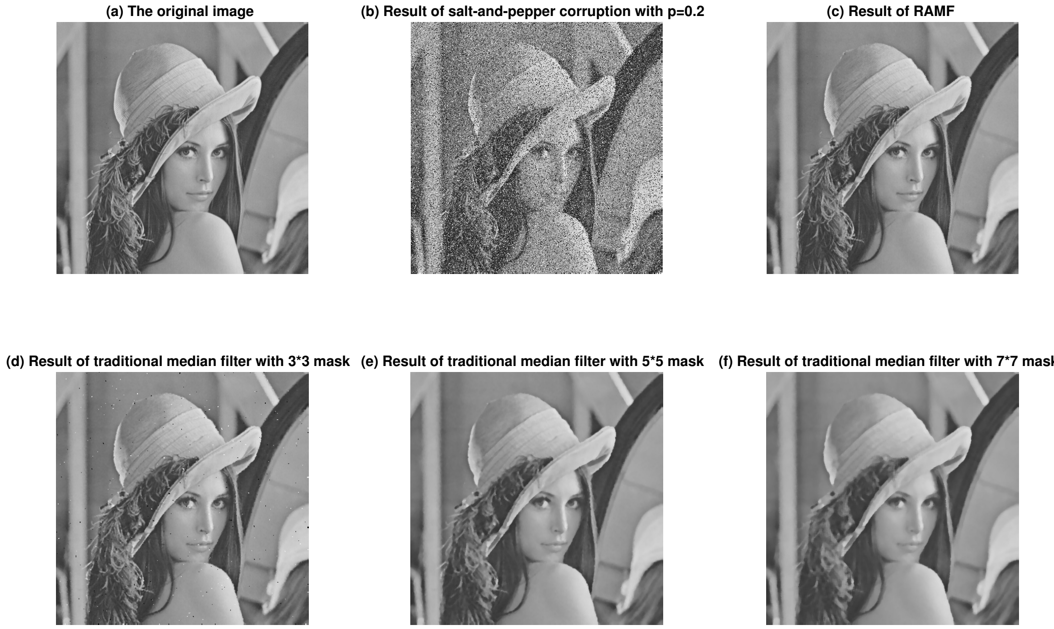


**Figure 3:** Comparison between RAMF and traditional median filters. Spatial density $p = 0.2$.

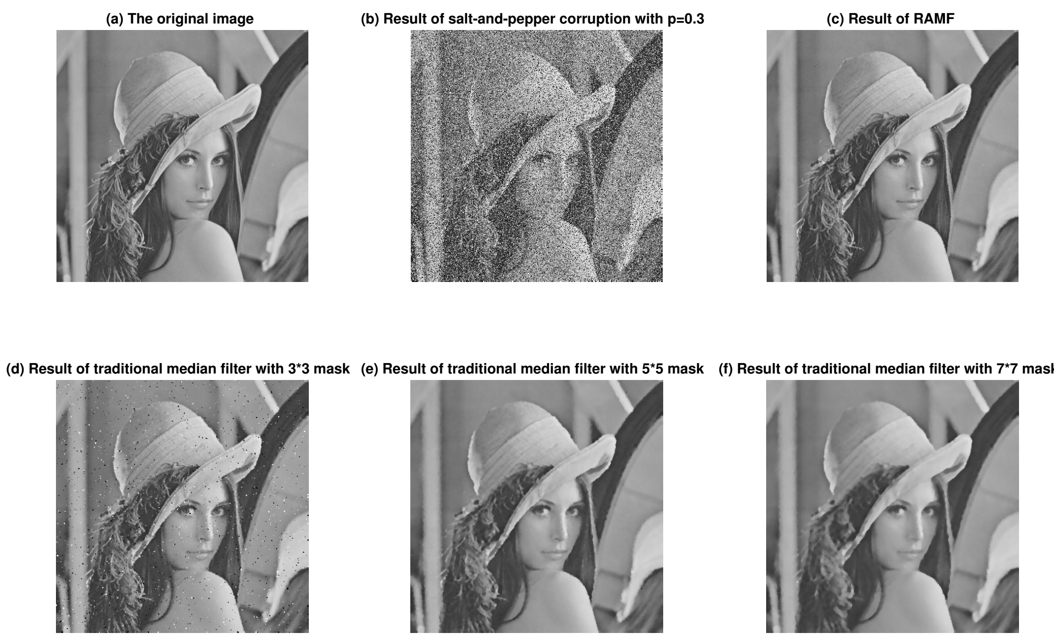


**Figure 4:** Comparison between RAMF and traditional median filters. Spatial density $p = 0.3$.

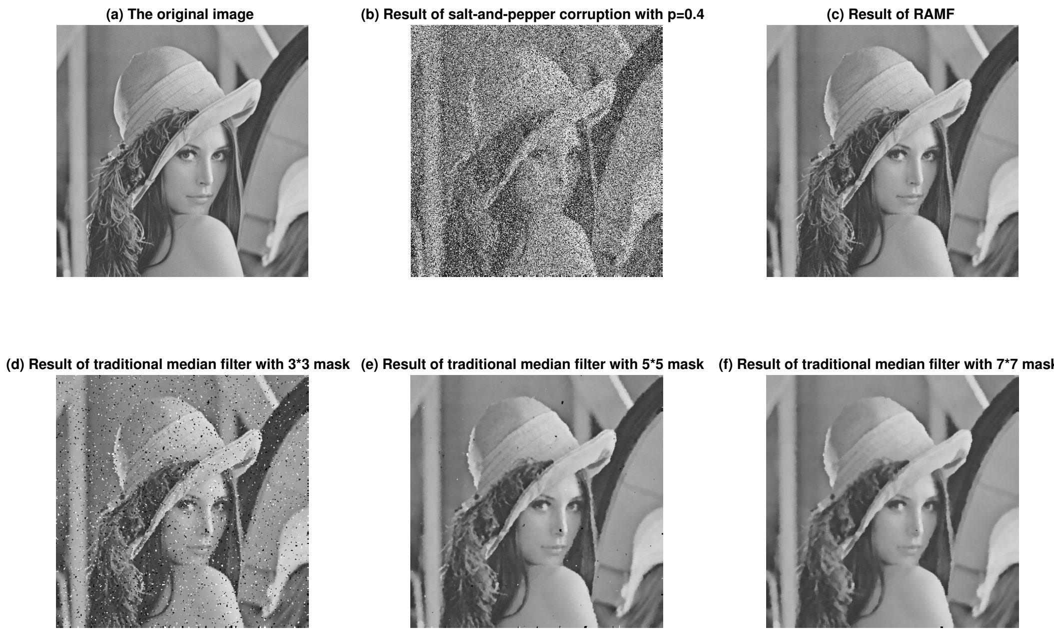


**Figure 5:** Comparison between RAMF and traditional median filters. Spatial density $p = 0.4$.

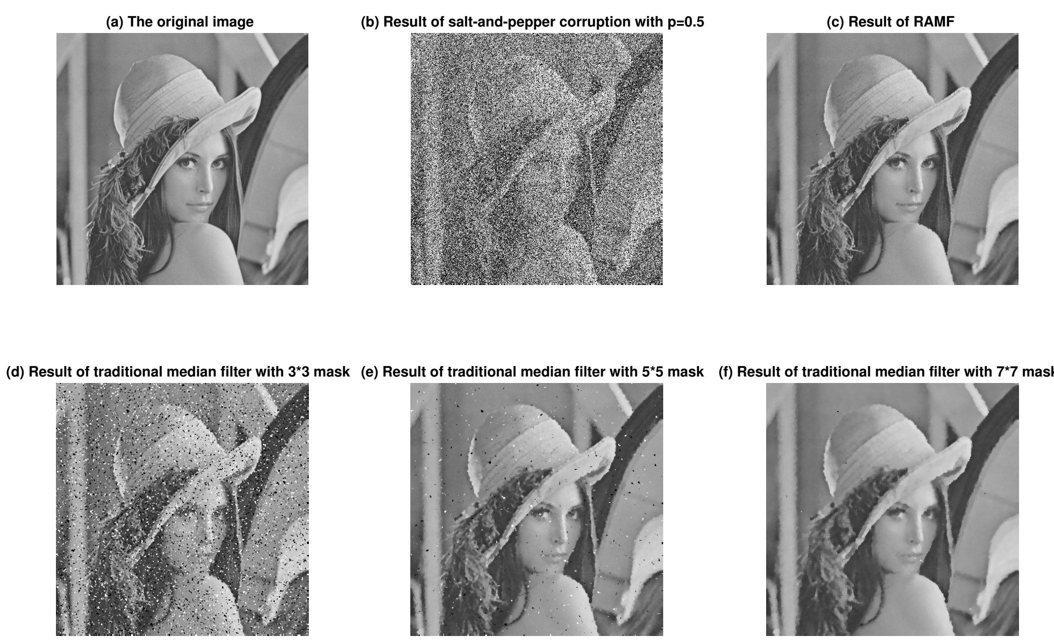


**Figure 6:** Comparison between RAMF and traditional median filters. Spatial density $p = 0.5$.

and nearly identical to the original image, regardless of the spatial density of impulses, as it both eliminates the noises and faithfully preserve the details of the original image.

# 4 Concluding Remarks

In this report, we implement a classic image restoration technique, the Ranked-order based Adaptive Median Filter, which is remarkably effective against images corrupted by salt-and-pepper noises. It can eliminate high-density impulses while preserving details of original image. Comparative experiments with traditional median filters with different sizes of masks vividly highlight its superiority. The whole project is insightful for newcomers in the field of digital image processing.

# References


[1] S Jayaraman, S Esakkirajan, and T Veerakumar. *Digital Image Processing*. Tata McGraw Hill Education, 2009.

[2] Humor Hwang and Richard A Haddad. Adaptive median filters: new algorithms and results. *IEEE Transactions on image processing*, 4(4):499–502, 1995.


# A Appendix

## A.1 Main Function: Frequency Analyzer

- *RAMF.m*.

```
%{
RAMF.m
The Ranked-order based Adaptive Median Filter Algorithm.
Zexi Huang
May.21 2017
%}
function outputImage=RAMF(inputImage,rmax)
%outputImage: the processed image of RAMF
%intputImage: the inputImage
%rmax: the maximal radius of the neighbourhood Sxy. Default value:5

%Default value of rmax
if nargin<2
    rmax=5;
end

%Initialize output image.
outputImage=inputImage;
outputImage(:)=0;

%The RAMF process.

```

```matlab
%Compute the zmin,zmax,zmed of the image with different radius.
[m,n]=size(inputImage);
zmin=zeros(rmax,m,n);
zmax=zeros(rmax,m,n);
zmed=zeros(rmax,m,n);
for r=1:rmax
    zmin(r,:,:)=ordfilt2(inputImage,1,ones(2*r+1,2*r+1),'symmetric');
    zmax(r,:,:)=ordfilt2(inputImage,(2*r+1)*(2*r+1),ones(2*r+1,2*r+1),'symmetric');
    zmed(r,:,:)=ordfilt2(inputImage,floor((2*r+1)^2/2),ones(2*r+1,2*r+1),'symmetric');
    %zmed(r)=medfilt2(inputImage,[2*r+1 2*r+1],'symmetric');
end

%Filter for each pixel.
for ii=1:m
    for jj=1:n
        r=1;
        loopFlag=1;
        while(loopFlag)
            if zmin(r,ii,jj)<zmed(r,ii,jj) && zmax(r,ii,jj)>zmed(r,ii,jj)
                if zmin(r,ii,jj)<inputImage(ii,jj) && zmax(r,ii,jj)>inputImage(ii,jj)
                    outputImage(ii,jj)=inputImage(ii,jj);
                    loopFlag=0;
                else
                    outputImage(ii,jj)=zmed(r,ii,jj);
                    loopFlag=0;
                end
            else
                if(r<rmax)
                    r=r+1;
                else
                    outputImage(ii,jj)=zmed(r,ii,jj);
                    loopFlag=0;
                end
            end
        end
    end
end
```

## A.2 Auxiliary Functions

- *ComparativeAnalysis.m.*

```
%{
ComparativeAnalysis.m
Generate comparative images processed by traditional median filter and
RAMF.
Zexi Huang
May.21 2017
%}

function ComparativeAnalysis(spatialDensity)
%spatialDensity: the spatial density of the sald−and−pepper noise

img=imread('lena.tiff');
corruptedImage=imnoise(img,'salt & pepper',spatialDensity);
subplot(2,3,1);
imshow(img);
title('(a) The original image');

subplot(2,3,2);
imshow(corruptedImage);
title(['(b) Result of salt−and−pepper corruption with p=',num2str(
    spatialDensity)]);

subplot(2,3,3);
RAMFoutputImage=RAMF(corruptedImage);
imshow(RAMFoutputImage);
title('(c) Result of RAMF');

subplot(2,3,4);
medianOutputImage33=medfilt2(corruptedImage,[3,3],'symmetric');
imshow(medianOutputImage33);
title('(d) Result of traditional median filter with 3*3 mask');

subplot(2,3,5);
medianOutputImage55=medfilt2(corruptedImage,[5,5],'symmetric');
imshow(medianOutputImage55);
title('(e) Result of traditional median filter with 5*5 mask');

subplot(2,3,6);
medianOutputImage77=medfilt2(corruptedImage,[7,7],'symmetric');
imshow(medianOutputImage77);
title('(f) Result of traditional median filter with 7*7 mask');
```