

# An Efficient Method for Removal of High-Density Salt-and-Pepper Noise

Xiaohui Guo

Chongqing College of Humanities, Science & Technology  
Chongqing Hechuan Grass Street Town

## ABSTRACT

In this paper, a novel algorithm for removing high-density salt-and-pepper noise from corrupted images is presented. Initially, salt-and-pepper pixels are identified in the detection stage. Then, the median value of non salt-and-pepper pixels is calculated, and the non salt-and-pepper pixels revolve around the current salt-and-pepper pixel in the  $3 \times 3$  window. The median value is the new grayscale values of the current salt-and-pepper pixel and it is calculated using the mean value. Simulation results indicate that the proposed method can remove high-density salt-and-pepper noise well and reserve image details, edges and textures best.

## General Terms

Remove high-density salt-and-pepper noise, Novel algorithm.

## Keywords

salt-and-pepper noise, identify, median value, mean value.

## 1. INTRODUCTION

We ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download the template, and replace the content with your own material.

Images are often influenced by various noises in the process of signal acquisition, transmission, reception or processing. For instance, it is inevitably affected by the external and internal interference due to the performance characteristic of transmission media and receiving equipment. So images might be corrupted by the impulse noise. And salt-and-pepper noise is a common impulse noise and the information of the original image is seriously damaged by salt-and-pepper noise. So removing salt-and-pepper noise is an important technical problem before subsequent image processing tasks such as edge detection or segmentation is carried out.

Many image denoising methods have been proposed to carry out salt-and-pepper noise suppression [1]. The median filter (MF) [2] is employed to implement the denoising process. MF is widely used due to computational convenience and edge preservation property, but it does not distinguish between noiseless and noise pixels, all pixels are replaced with neighborhood median pixels [3], in addition to this, it is quite incompetence for removing high-density salt-and-pepper noise. In order to overcome the algorithm's malpractice, many scholars have explored much.

In [4], a novel two-stage noise adaptive fuzzy switching median (NAFSM) filter for effective removal of salt and pepper noise was presented. The histogram of the corrupted image is utilized to identify noise pixels in the detection stage. Usually, when the gray value of pixels is 0 or 255 for an 8-bit gray level images, the pixels are defined as salt and pepper pixels. NAFSM is able to suppress high-density of salt-and-pepper noise very well, NAFSM is able to suppress high-

density of salt-and-pepper noise very well, but the threshold parameter need to be set manually.

In [5], one efficient and fast method for the removal of high-density salt-and-pepper noise (EFRN) was presented. Initially, salt-and-pepper noise pixels are identified by using the histogram of the corrupted image [4]. Then, the mean value of non salt-and-pepper pixels was calculated in the  $3 \times 3$  window centered at the current salt-and-pepper pixel. The algorithm is able to suppress high-density of salt-and-pepper noise well and the processing time is less than NAFSM filter. However, it is not able to preserve image details, edges and textures well.

In this paper, a novel algorithm for removing high-density salt-and-pepper noise from corrupted images is presented. Initially, salt-and-pepper pixels are identified in the detection stage. Then, the median value of non salt-and-pepper pixels is calculated in the  $3 \times 3$  window centered at the current salt-and-pepper pixel. The median value is the new grayscale values of the current salt-and-pepper pixel and it is calculated by using the mean value. Our method is able to suppress high-density of salt-and-pepper noise and preserve image details, edges and textures well. In addition, in order to reduce the time compared to the sorting method, the mean value is used to calculate the median value.

The rest of this letter is organized as follows. In Section II, the proposed algorithm is introduced. The implementation results and comparison are provided in Section III and Section IV concludes the work in this paper.

## 2. OUR PROPOSED METHOD

The proposed algorithm consists of two stages: detection stage and filtering stage. The former determines which salt-and-pepper pixels are identified by using the histogram of the corrupted image [4]. When pixels are detected as salt-and-pepper pixels, these pixels will enter into the second phase – the filtering stage. In the filtering stage, the median value of non salt-and-pepper pixels is calculated, and the non salt-and-pepper pixels revolve around the current salt-and-pepper pixel in the  $3 \times 3$  window. The median value is the new grayscale values of the current salt-and-pepper pixel and it is calculated by using the mean value. But, if pixels are detected as noise-free pixels, these pixels are retained and left unprocessed.

### A. Detection Stage

In the detection stage, salt-and-pepper noise pixels are identified by using the histogram of the corrupted image [4]. If the gray value of pixels is 0 or 255 for an 8-bit gray level images, the pixels are defined as salt and pepper pixels.

For image  $X$ , a binary noise mask  $F_{ij}$  will be created to mark the location of salt-and-pepper pixels by using

$$F_{ij} = \begin{cases} 1, & x_{ij} = 0 \text{ or } 255 \\ 0, & x_{ij} \neq 0 \text{ or } 255 \end{cases} \quad (1)$$

where  $x_{ij}$  is the pixel at location  $(i, j)$  with image  $X$ . When the gray value of  $x_{ij}$  is 0 or 255,  $F_{ij} = 1$ . This shows that  $x_{ij}$  represents salt-and-pepper pixel, otherwise  $F_{ij} = 0$ . This shows that  $x_{ij}$  represents noise-free pixel.

### B. Filtering Stage

For each pixel in image  $X$ , a  $3 \times 3$  window is defined first. Let  $S_{ij}$  represent the set of pixels revolve around the pixel  $x_{ij}$  in the  $3 \times 3$  window. Thus, it can be given as

$$S_{ij} = \{(k, l) \mid i - 1 \leq k \leq i + 1, j - 1 \leq l \leq j + 1\} \quad (2)$$

In order to prevent salt-and-pepper pixels to be processed are influenced by currently processed noise pixels, the image  $Y$  is defined. And each pixel value of image  $Y$  is given as follows:

$$y_{ij} = \begin{cases} m_j, & F_{ij} = 1 \\ x_{ij}, & F_{ij} = 0 \end{cases} \quad (3)$$

where  $F_{ij}$  is a binary noise mask,  $m_j$  is the median value of the gliding window,  $x_{ij}$  is the pixel at location  $(i, j)$  of the original image  $X$ ,  $y_{ij} = m_j$  means the gray value of salt-and-pepper pixel is replaced by the median value, at the same time, the value of  $F_{ij}$  is set to 0, and  $y_{ij} = x_{ij}$  implies the pixel gray-level remains unchanged.

Let  $M_j$  represents other noise-free pixels revolve around the pixel  $x_{ij}$  in the  $3 \times 3$  window. Thus, it can be given as

$$M_j = \{y_{k,l} \mid y_{k,l} \in S_{ij}, (k, l) \neq (i, j), F_{k,l} = 0\} \quad (4)$$

where  $M_j = 0$  implies the median value do not need to be calculated.

In order to reduce the processing time, the computation of the median value will utilize the mean value rather than the sorting method. Let Set  $A$  represents  $M_j$ ,  $m$  means the median value. The detailed steps of computation of the median value are described as follows.

1)First, the number of elements of Set  $A$  is evaluated. If the number is 1, the element of Set  $A$  is  $m$ ; otherwise, jump to Step 2.

2)The mean value of elements of Set  $A$  is calculated, and  $M$  means the mean value.

3)The elements of Set  $A$  are divided by Set  $H$  and Set  $L$  by  $M$ . Where Set  $H$  is a set greater than or equal to  $M$ , and Set  $L$  is less than  $M$ .

4)The number of Set  $H$  and Set  $L$  are calculated, and the result are  $N_H$  and  $N_L$ , respectively.

5)  $N_L$  is evaluated. If  $N_L = 0$ , the first element of Set  $H$  is  $m$ , and it means all the elements of Set  $A$  are the same; otherwise, jump to Step 6.

6)The size of  $N_H$  and  $N_L$  are evaluated. If  $N_L \geq N_H$ , the a'th element is obtained from small to large in Set  $L$  by using the sorting method and it is  $m$ , in addition, the value of  $a$  is  $\lceil N$

$\lceil \frac{N}{2} - N_L + 1 \rceil$ ; otherwise, the b'th element is obtained from small to large in Set  $H$  by using the sorting method and it is  $m$ , in addition, the value of  $b$  is  $\lceil \frac{N}{2} - N_L + 1 \rceil$ .

## 3. EXPERIMENTAL RESULTS

In this section, our method will be compared with EFRN [5] based on their experimental results. To verify the characteristics and performances of the two denoising algorithms, three images of  $512 \times 512$  8-bit gray-scale are chosen for the test. The test images include "Lena," "Baboon" and "Jet". These images are corrupted by salt-and-pepper noise ranging from 10% to 90% with increment steps of 10%, and these noise pollution images are tested.

The peak signal-to-noise ratio (PSNR) is used to assess the quantitative quality of the reconstructed images. And when the value of PSNR is bigger, the quality of the reconstructed image is better. Table I, Table II and Table III list the restoration results in PSNR (dB) for image Lena, Baboon and Jet with different noise ratios, respectively. Obviously, the experimental results of our method and EFRN are similar in PSNR.

**Table 1. COMPARISONS OF RESTORATION RESULTS IN PSNR (dB) FOR IMAGE "LENA" WITH DIFFERENT NOISE RATIOS**

Noise	PSNR for image 'Lena' (dB)	
	EFRN	Our method
10%	32.08	32.66
20%	29.88	30.46
30%	28.13	28.69
40%	26.79	27.36
50%	25.82	26.19
60%	24.37	24.95
70%	22.70	23.04
80%	18.77	19.05
90%	13.07	12.34

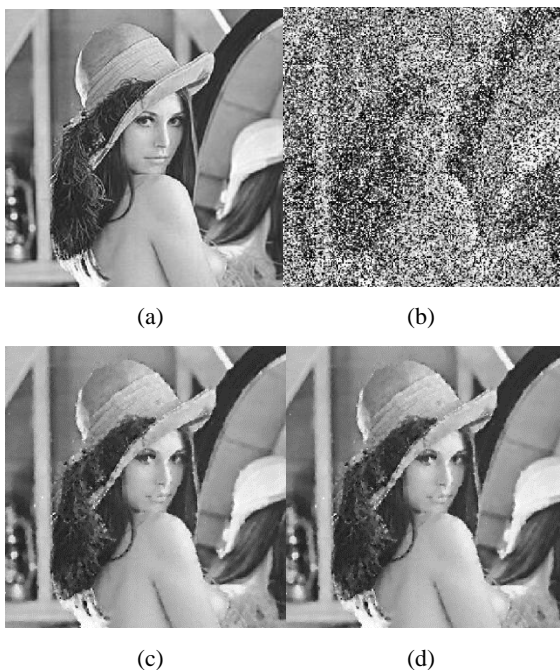
**Table 2. COMPARISONS OF RESTORATION RESULTS IN PSNR (dB) FOR IMAGE "BABOON" WITH DIFFERENT NOISE RATIOS**

Noise	PSNR for image 'Baboon' (dB)	
	EFRN	Our method
10%	31.87	31.57
20%	28.82	28.53
30%	26.94	26.59
40%	25.50	25.22

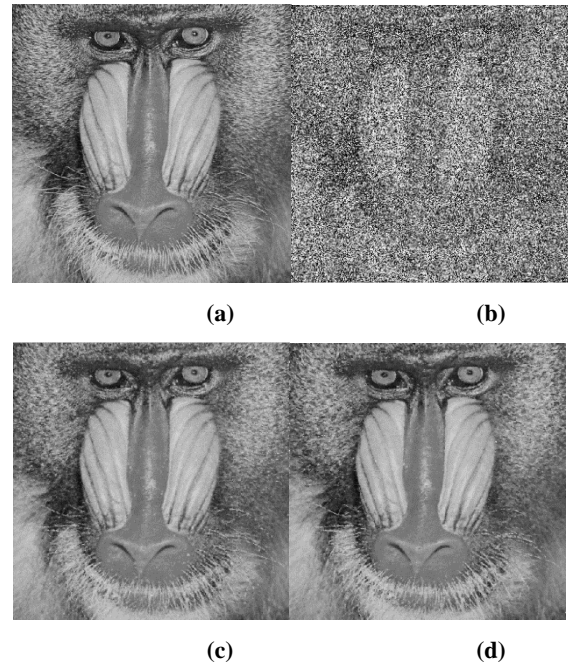
50%	24.22	24.08
60%	23.02	22.95
70%	21.39	21.48
80%	18.32	18.20
90%	12.78	12.19

**Table 3. COMPARISONS OF RESTORATION RESULTS IN PSNR (dB) FOR IMAGE “JET” WITH DIFFERENT NOISE RATIOS**

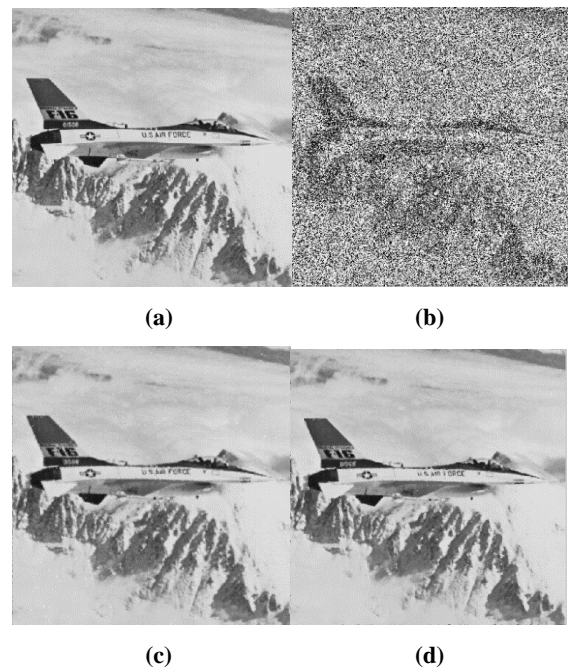
Noise	PSNR for image ‘Jet’ (dB)	
	EFRN	Our method
10%	39.15	38.82
20%	35.74	35.60
30%	33.63	33.24
40%	32.16	31.70
50%	30.77	30.47
60%	29.30	29.15
70%	26.86	27.07
80%	22.62	22.66
90%	16.57	15.81



**Fig.1. Experimental results of different methods in restoring corrupted image“Lena.” (a) Original noise-free image, (b) corrupted image with 60% salt-and-pepper noise, (c) EFRN, and (d)our method.**

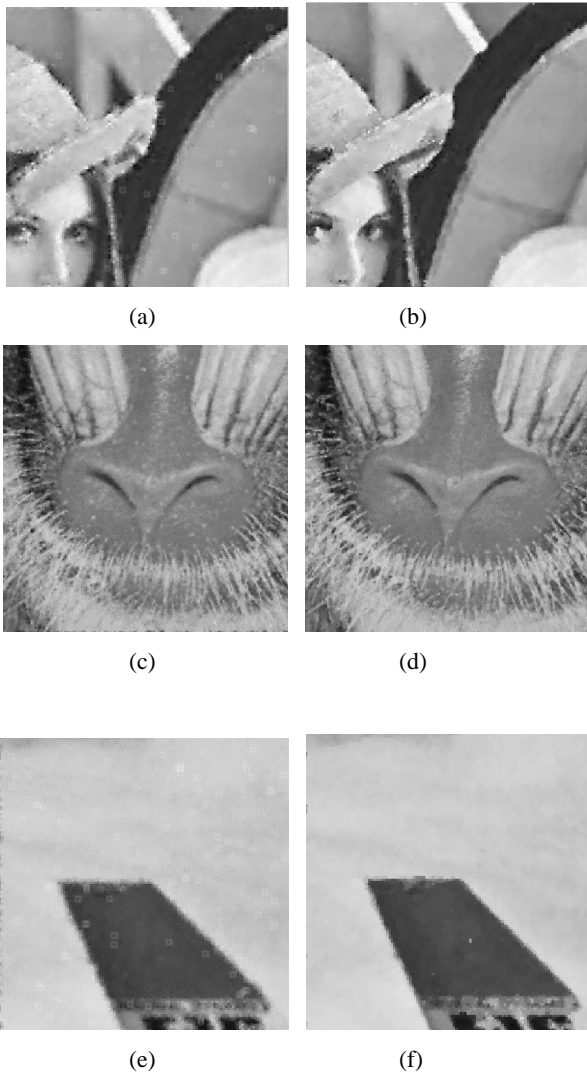


**Fig.2. Experimental results of different methods in restoring corrupted image“Baboon.” (a) Original noise-free image, (b) corrupted image with 60% salt-and-pepper noise, (c) EFRN, and (d)our method.**



**Fig.3. Experimental results of different methods in restoring corrupted image“Jet.” (a) Original noise-free image, (b) corrupted image with 60% salt-and-pepper noise, (c) EFRN, and (d) our method.**

Fig. 1, Fig. 2 and Fig. 3 show the reconstructed images of “Lena”, “Baboon” and “Jet” for different denoising methods with a noise ratio of 60%, respectively. In order to explore the visual quality, Fig. 4 shows the local features of the “Lena” image, the “Baboon” image and the “Jet” image for the two filters with a noise ratio of 60%. And the figure shows that our algorithm obtains the best results and reserve image details, edges and textures best.



**Fig.4. Comparison of local features for images (Lena, Baboon and Jet) with noise ratio of 60% with two methods. The image is (a) EFRN (Lena). (b) Our method (Lena). (c) EFRN (Baboon). (d) Our method (Baboon). (e) EFRN (Jet). (f) Our method (Jet).**

This paper proposes a novel algorithm for removing high density salt-and-pepper noise from corrupted images. In the first stage, salt-and-pepper pixels are identified. In the second stage, the median value of non salt-and-pepper pixels is calculated, and the non salt-and-pepper pixels revolve around the current salt-and-pepper pixel in the  $3 \times 3$  window. And the median value is the new grayscale values of the current salt-and-pepper and it is calculated using the mean value. The experimental results show that our method and EFRN are similar in PSNR. But, the local features obtained by these two methods, it is discovered that our method obtains the best results and reserve image details, edges and textures best.

#### 4. REFERENCES

- [1] Chen, Pei-Yin, and Chih-Yuan Lien. "An efficient edge-preserving algorithm for removal of salt-and-pepper noise." *IEEE Signal Processing Letters* 15 (2008): 833-836.
- [2] Nodes, Thomas, and Neal Gallagher. "Median filters: Some modifications and their properties." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 30.5 (1982): 739-746.
- [3] Wang, Yi, et al. "An Efficient Adaptive Fuzzy Switching Weighted Mean Filter for Salt-and-Pepper Noise Removal." *IEEE Signal Processing Letters* 23.11 (2016): 1582-1586.
- [4] Toh, Kenny Kal Vin, and Nor Ashidi Mat Isa. "Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction." *IEEE signal processing letters* 17.3 (2010): 281-284.
- [5] Zong-wei, Lyu, et al. "One Efficient and Fast Method for the Removal of High-Density Salt and Pepper Noise [J]." *Acta Electronica Sinica* 8 (2011): 032.
- [6] Zhang, L. I., et al. "Mean-based fast median filter." *JOURNAL-TSINGHUA UNIVERSITY* 44.9 (2004): 1157-1159.
- [7] Bao, Hua, et al. "Fast median filtering algorithm based on mean searching." *Journal of Sichuan University: Engineering Science Edition* (2011): 76-77.