An Efficient Procedure for Removing Salt and Pepper Noise in Images

Guangyu Xu and Muhammad Jibril Aminu
E-mail: xgy761220@163.com, gyxu@aust.edu.cn
School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan, China

Keywords: noise removal, salt and pepper noise, edge preservation, extremum-compressing median filtering

Received: January 10, 2022

In this paper, we propose an efficient algorithm for removing salt and pepper noise in images. The process of denoising is implemented in two stages: noise detection followed by noise removal. For noise detection, two extreme intensity values in an image are used to detect possible "noise pixels". For noise removal, the switching mechanism only selects "noise pixels" for processing to avoid altering any fine image details, and only the identified noise-free pixels are used to achieve better denoising performance. Two filtering techniques, the edge-preserving filtering (EPF) and the extremum-compressing median filtering (ECMF), are employed for edge-preserving and noise removal. The EPF provides higher correlation between the corrupted pixel and neighborhood pixel, which gives rise to better edge preservation. The ECMF can yield an appropriate estimation by selecting the median pixel from the noise-free pixels of current filtering window. The proposed algorithm is tested on different images and provides a better restoration performance over some of the salt and pepper noise filters.

Povzetek: V prispevku je predstavljena učinkovita metoda za odstranjevanje šuma iz slik.

1 Introduction

Digital images are often corrupted by salt and pepper noise in the process of image acquisition and transmission. Salt and pepper noise is a special case of impulse noise, where a certain percentage of pixels in the image are randomly digitized into two extreme intensities. Normally, these intensities have the maximum and minimum intensities within the dynamic range. It is very important to remove such noise before subsequent image processing tasks such as edge detection, segmentation or object recognition is carried out, because the occurrence of salt and pepper noise can severely damage the information or data embedded in the original image. To this end, a variety of techniques have been proposed for removal of salt and pepper noise.

The standard median filter (MF) [1] was once the most popular filter for removing impulse noise because of its good denoising power and computational efficiency. However, since each pixel in the image is replaced by median value in its neighborhood, the median filter often removes desirable details and blurs it, too. The weighted median filter [2] and the center-weighted median filter [3] were proposed to improve the MF by giving more weight to some selected pixels in the filtering window. Unfortunately, these two filters are still carried out uniformly across the image without considering whether the current pixel is noise free or not. These filters are effective only for low noise densities.

Over the years, better noise removal methods with different kinds of noise detectors have been proposed, for example, switching median filter (SMF) [4], adaptive median filter (AMF) [5], tri-state median filter (TSMF) [6], adaptive center-weighted median filter (ACWMF) [7], decision-based algorithm (DBA) [8], [9], a two-stage filter [10], etc. The basic idea of the methods is that the noisy pixels are detected first and eliminated afterward, whereas the uncorrupted pixels are left unchanged to prevent the blurred or removal of fine details. The main drawback of these filters is that they only use median values or their variations to restore the noisy pixels without considering local features such as the possible presence of edges, and hence they cannot usually preserve image details even when the images are corrupted by low noise densities. To overcome the above drawbacks, an efficient edge-preserving algorithm (EEPA) in [11] was introduced for removal of salt and pepper noise without degrading image fine details. It only performs well when an image is corrupted by 50% salt and pepper noise or lower. Kenny and Nor in [12] proposed a noise adaptive fuzzy switching median filter (NAFSMF) for removal of salt-and-pepper noise. This method can suppress high density of noise, at the same time preserving fine image details. Similarly, an efficient adaptive fuzzy switching weighted mean filter was presented in [13] for salt and pepper noise removal. In [14], an adaptive type-2 fuzzy method was introduced for eliminating salt and pepper noise.

Apart from the median-based filters, some other types of methods were also presented. The edge-preserving regularization method [15] and total variation L1 regularization [16] were applied to the noisy pixels to preserve the edges and noise suppression. These two methods are shown to be most efficient in dealing with high-density of salt-and-pepper noise, meanwhile preserving image edges. In [17], the cardinal B-splines was used to restore the noisy pixels by selecting four nearly noise-free pixels. A novel filter based on the continued fractions interpolation was introduced in [18] for removal of salt-and-pepper noise. Then, Ramadan [19] used the simple average filtering for the noise suppression and edge-preserving; however, it is unsatisfactory in restoring details and edges, especially when the noise density is high. Moreover, the weighted average based filtering strategy has successfully been applied in removing salt and pepper noise [13], [20]. In [21], a universal impulse noise filter was proposed to remove any type of impulse noise by combining the noise detection with the nonlocal means filter.

In this paper, an efficient algorithm is proposed for removing salt-and-pepper noise from noisy images. The process of denoising is carried out in two stages: noise detection followed by noise removal. For noise detection, the noisy pixels are detected, based on the fact that their values must be the extreme intensities of the image. For noise removal, the switching mechanism only selects "noise pixels" for processing to avoid altering any fine image details, and only the identified noise-free pixels are used for noise removal to achieve better denoising performance. The proposed algorithm employs two different filtering techniques for edge-preserving and noise removal, namely edge-preserving filtering (EPF) and extremum-compressing median filtering (ECMF). The EPF adopts a directional correlation filtering technique based on observing the sample correlations of four different directions. Higher correlation gives rise to better edge preservation. The ECMF can yield an appropriate estimation of the processed pixel by selecting the median pixel from the noise-free pixels in the current filtering window. Extensive simulations show that the proposed algorithm not only can provide better performance of suppressing salt and pepper noise, but also can preserve more details. Furthermore, our method produces good restoration results in filtering color images.

The rest of the paper is organized as follows. In Section 2, noise model is introduced. The proposed algorithm is presented in Section 3. Experimental results are given in Section 4. Section 5 concludes the work in this paper.

2 Noise model

When an image is corrupted by salt and pepper noise, a portion of the pixel is changed with random values. To be precise, let $x_{i,j}$ and $u_{i,j}$ denote the intensity values of a noise-free image and the noisy image at the pixel location (i,j). Let the dynamic range of the image be $[L_{min}, L_{max}]$. If the noise ratio is p, then

$$u_{i,j} = \begin{cases} n_{i,j}, & with\ probability\ p \\ x_{i,j}, & with\ probability\ 1-p \end{cases} \tag{1}$$

where $n_{i,j}$ is the intensity value of the noisy pixel in noisy image u. There are two models of impulse noise: the salt and pepper noise where $n_{i,j}$ is equal to L_{min} or L_{max}, and the random-valued impulse noise where $n_{i,j}$ takes random values from the interval $[L_{min}, L_{max}]$ with a uniform distribution. In this paper, we only focus on the removal of salt and pepper noise.

3 Proposed algorithm

The proposed algorithm is a two-stage filter for salt and pepper noise detection and removal. Initially, the detection stage utilizes two extreme intensity values to identify possible "noise pixels". These detected "noise pixels" will be subjected to the following filtering action, while noise-free pixels are left unchanged. Then, two different filtering techniques, namely the EPF and the ECMF, are employed for edge-preserving and noise removal.

3.1 Noise detection

Noise detection is based on the fact that a digital image corrupted with salt and pepper noise would produce two extreme intensity values, the minimal intensity value L_{min} and the maximal intensity value L_{max}. One can identify these two intensities by analyzing the noisy image histogram, where the isolated peaks clearly indicate the two salt and pepper noise intensities. Normally, for 8-bit grayscale images corrupted by salt and pepper noise, noisy pixel only takes either L_{min} or L_{max} as its intensity value. Therefore, these two noise intensities will be used to identify possible "noise pixels" in the images. A binary noise map $F_{i,j}$ will be created to mark the location of "noise pixels" by using

$$F_{i,j} = \begin{cases} 0, & u_{i,j} = L_{\min}\ or\ L_{\max} \\ 1, & otherwise \end{cases} \tag{2}$$

where $F_{i,j} = 1$ represents noise-free pixels to be retained from the noisy image, whereas $F_{i,j} = 0$ represents "noise pixels" subjected to the next filtering process.

Since the detection of "noise pixels" is based on L_{min} and L_{max}, noise-free pixels may be falsely identified as "noise pixels" at image uniform regions having same values as L_{min} or L_{max}. In this case, it is difficult for the filter to determine the size of the filtering window. On the other hand, when the filtering action is applied these pixels, their values may be changed. In [19], the intensity values 0 and 255 are employed to identify noise pixels, but the author does not provide the solution on how to prevent the above problems from occurrence. In the section 3.3, we describe in detail how to deal with this problem.

3.2 Edge-preserving filtering (EPF)

After the binary noise map $F_{i,j}$ is created, noise-free pixels marked with $F_{i,j} = 1$ will be retained and the filtering action is omitted to avoid altering any details in the original images, whereas "noise pixels" marked with $F_{i,j} = 0$ will be replaced by an estimated correction term.

Before introducing the EPF, we review a basic fact, that is, a noise-free image consists of locally smoothly varying areas separated by edges. Considering this characteristic, the proposed EPF adopts a directional correlation-dependent filtering technique based on observing the sample correlations of four main directions. Here, we only focus on the edges aligned with four main directions. The proposed algorithm uses a $(2N+1) \times (2N+1)$ filtering window $W_{i,j}^N$, given as

$$W_{i,j}^N = \{u_{i+s,j+t}\} where s,t \in (-N, \cdots, 0, \cdots, N). \quad (3)$$

Then, the number of noise-free pixels, $G_{i,j}^N$ in the window $W_{i,j}^N$, is counted using

$$G_{i,j}^N = \sum_{s,t \in (-N,\cdots,0,\cdots,N)} F_{i+s,j+t}. \quad (4)$$

For each "noise pixel" marked with $F_{i,j} = 0$, the EPF will detect edges in its four directions in the 3×3 filtering window, i.e., $W_{i,j}^1$. For simpler representation, let $a,\ b,\ c,\ d,\ e,\ f,\ g,$ and h represent those pixel values, $u_{i-1,j-1}$, $u_{i-1,j}$, $u_{i-1,j+1}$, $u_{i,j-1}$, $u_{i,j+1}$, $u_{i+1,j-1}$, $u_{i+1,j}$, and $u_{i+1,j+1}$, respectively, around the current pixel $u_{i,j}$ as shown in Figure 1. The detailed steps of the proposed EPF are described as follows.

If the current filtering window $W_{i,j}^1$ has at least two noise-free pixels, i.e., $G_{i,j}^1 > 1$, do the following.

Step 1: Calculate the four directional differences around the pixel $u_{i,j}$ in the $W_{i,j}^1$.

$$\begin{cases} D_1 = |a-h|, & D_2 = |b-g| \\ D_3 = |c-f|, & D_4 = |d-e| \end{cases} \quad (5)$$

Step 2: Check whether the pixels $(a,\ b,\ c,\ d,\ e,\ f,\ g,$ and $h)$ are possible "noise pixels" marked as $F_{i,j} = 0$, respectively. If yes, the pixel might be corrupted, and thus we do not consider the directional differences containing it by setting those differences to 512.

Step 3: Find the minimum value among four directional differences and denote it as D_{min}. The minimum directional difference has the strongest correlation and probably has an edge in its direction. Hence, the restored value $y_{i,j}$ of the corrupted pixel $u_{i,j}$ is estimated as follows:

$$y_{i,j} = \begin{cases} \frac{(a+h)}{2,} & if D_{min} = D_1 \\ \frac{(b+g)}{2,} & if D_{min} = D_2 \\ \frac{(c+f)}{2,} & if D_{min} = D_3 \\ \frac{(d+e)}{2,} & if D_{min} = D_4 \end{cases} \quad (6)$$

Note that the EPF is used only when $G_{i,j}^1 > 1$, here the filtering window size is 3×3. The reason is that there is lower directional correlation between the central pixel and its neighbors which are spatially far away from the central pixel.

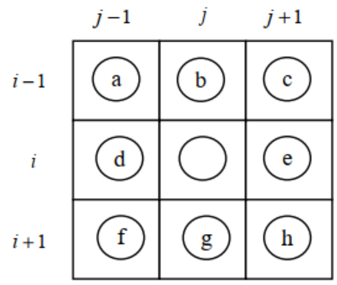

Figure 1: The pixels around the current pixel.

3.3 Extremum-compressing median filtering (ECMF)

Obviously, the EPF can prevent the noise-free pixels from being changed and give better estimates of the noisy pixels than median alone. However, there is an exception in the EPF. If D_{min} is equal to 512, it means that there exist the corrupted pixels on four directions in the window $W_{i,j}^1$. In this condition, no edge is considered. Here, noisy pixels marked with $F_{i,j} = 0$ are excluded and not involved in the estimate of the currently processed pixel. That is, only those noise-free pixels marked with $F_{i,j} = 1$ are used as candidates for selecting the median pixel, $y_{i,j}$, given by

$$y_{i,j} = median\{u_{i+s,j+t}\} \ \ with \ \ F_{i+s,j+t} = 1. \quad (7)$$

On the other hand, if the current filtering window $W_{i,j}^1$ does not have a minimum number of one noise-free pixel (i.e., $G_{i,j}^1 = 0$), then the filtering window will be expanded by one pixel at each of its four sides. This procedure is repeated until the criterion of $G_{i,j}^N \geq 1$ is met, and then the filtering action is applied to the current pixel. For example, when $G_{i,j}^2 \geq 1$ (i.e., the window $W_{i,j}^2$ contains noise-free pixels), the median pixel in the $W_{i,j}^2$ will be selected as the estimate of the current pixel $u_{i,j}$ by using (7). Accordingly, if $G_{i,j}^3 \geq 1$, the median filtering based on (7) is applied to the window $W_{i,j}^3$. This strategy of choosing only noise-free pixels is imposed to avoid selecting a "noise pixel" as the estimated median.

As mentioned above, the noise detection based on L_{min} and L_{max} may falsely identify noise-free pixels as "noise pixels" at image uniform regions having same intensities as L_{min} or L_{max}. Consequently, the filtering window will be expanded continuously and the selected median pixel may be improper to be used as an estimate term. Considering this possibility, the search for "noise-free pixels" is halted when the size of the window has reached 7×7 (or $N = 3$) although no "noise-free pixel" is marked, i.e., $G_{i,j}^3 = 0$. Here, the window $W_{i,j}^1$ centered at (i,j) will be used to estimate the median pixel $y_{i,j}$, i.e.,

$$y_{i,j} = median\{u_{i+s,j+t}\} \ \ where \ \ s,t \in (-1,0,1). \quad (8)$$

By using (8), the proposed algorithm can avoid modifying uncorrupted pixels at image uniform regions having same intensities as L_{min} or L_{max}, meanwhile removing corrupted pixels. We call the above approach as the ECMF. However, it would remove the isolated noise-free pixels with intensity L_{min} or L_{max}. To the best of our knowledge, this is still an unsolved problem.

4 Experimental results and discussions

4.1 Experimental environment

In this section, we compare our method with a number of existing filters. To verify the characteristics and performances of various filters, a variety of simulations are carried out on six 8-bit gray-scale test images: Lena, Boats, Peppers, Goldhill, Barbara, and Rice. In the simulations, images are corrupted with salt and pepper noise, where 0 represents the “pepper” noise and 255 represents the “salt” noise with equal probability. A wide range of noise ratios varied from 10% to 90% with increments of 10% is tested. Totally, five recent filters are compared with our method in terms of objective testing (quantitative evaluation) and subjective testing (visual quality): AMF [5], ACWMF [7], DBA [8], EEPA [11], NAFSMF [12]. In each experiment, the parameters or thresholds of the compared methods are set as suggested by its authors. To be fair, for high-density of noise, AMF, ACWMF, and EEPA filters are iteratively executed to obtain the best results.

The experiments are implemented in Matlab R2018b on PC equipped with an Intel Core i7-8700 3.20GHz, 4.0GB RAM, and Windows 10. We employ the peak signal-to-noise ratio (PSNR) to assess the quantitative quality of the restored images for various methods. Higher PSNR values indicate better image restoration.

4.2 Experimental results

The comparison of restoration results in PSNR for six test images (Top to bottom and left to right: Lena, Boats, Peppers, Goldhill, Barbara, and Rice) corrupted with various salt and pepper noise ratios are shown in Figure 2. It is easy to see that our method is very good in removing salt-and-pepper noise and has the highest PSNR values at all levels of noise. In Table 1, we list the restoration results in PSNR of different filters for images “Lena” and “Boats” corrupted with salt and pepper noise. Apparently, our method provides the best results than the others.

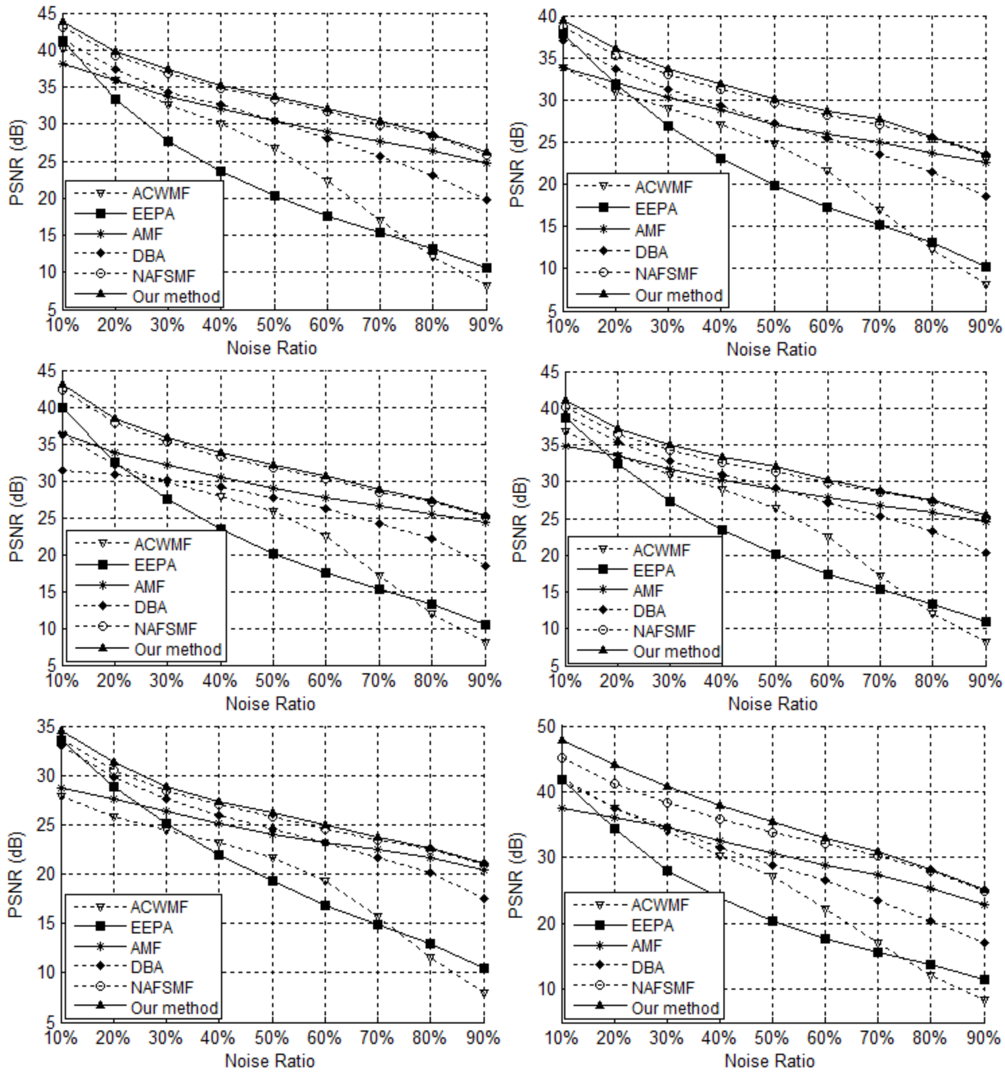

Figure 2: Comparison of restoration results in PSNR.

Filters	Lena			Boats		
	30%	70%	90%	30%	70%	90%
ACWMF	32.83	17.09	8.15	28.96	16.93	8.05
EEPA	27.74	15.31	10.53	26.97	15.18	10.27
AMF	33.66	27.75	24.80	30.36	24.92	22.65
DBA	34.31	25.72	19.86	31.31	23.50	18.63
NAFSMF	36.85	29.95	25.84	32.99	27.04	23.34
Our method	**37.43**	**30.37**	**26.14**	**33.64**	**27.67**	**23.54**

Table 1: Comparisons of restoration results in PSNR.

Furthermore, a subjective visual result of the noise removal is presented in Figure 3. Figure 3(b) is the noisy "Boats" image with 70% salt and pepper noise. Among the restorations, the ACWMF and EEPA have plenty of visible noise patches and produce disappointing results, whereas the AMF, DBA, NAFSMF, and our method work well to remove noise. One can observe that our method gives the best performance in terms of noise removal and edge preservation such as ropes between masts.

In Figure 4, we show the restoration results of different filters in restoring image "Lena" corrupted with 90% salt and pepper noise. Obviously, only the NAFSMF and our method can produce perceptible reconstructed images. However, our method has a better noise suppression ability as compared to the NAFSMF since the restored images does not have any spot of unremoved noise. Moreover, our method can preserve edges better. The blurring caused by the NAFSMF is mainly due to its median-based restoration mechanism. The DBA can suppress noise satisfactorily, but at the expense of image details, causing edges to be jagged and distorted. The AMF is capable of removing noise, but it blurs image details significantly. On the other hand, the ACWMF and EEPA have completely failed to remove noise.

The average processing time takes by the filters at various noise densities is shown in Figure 5, where a total of six test images are processed. Due to the computational complexity of the edge-preserving filtering is high, our method has a higher processing time than the AMF and DBA, but in general it is lower than the ACWMF, EEPA, and NAFSMF. However, the higher processing time is compensated by the better restoration results as shown above.

4.3 Application on color images

In this paper, the RGB color space is selected to represent the color images. The noisy color images are obtained by applying the noise model in (1) to each of the R-, G-, and B- channels independently. This means when a color image is corrupted by salt and pepper noise with noise ratio p, then each color component image is being corrupted with p. Accordingly, our proposed filter can be extended for removing noise from corrupted color images by applying the proposed algorithm to the R-, G-, and B- channels independently.

Simulation results for color image "Lena" corrupted with 80% salt and pepper noise are depicted in Figure 6. It is observed that our proposed mehtod exhibits excellent noise suppression performances, meanwhile preserving image fine details well.

5 Conclusion

An efficient algorithm for the removal of salt-and-pepper noise is presented. To remove the salt-and-pepper noise with edge and fine detail preservation, the switching mechanism is used to avoid altering any fine details in the images, and then the EPF provides more edge details, leading to better edge preservation.

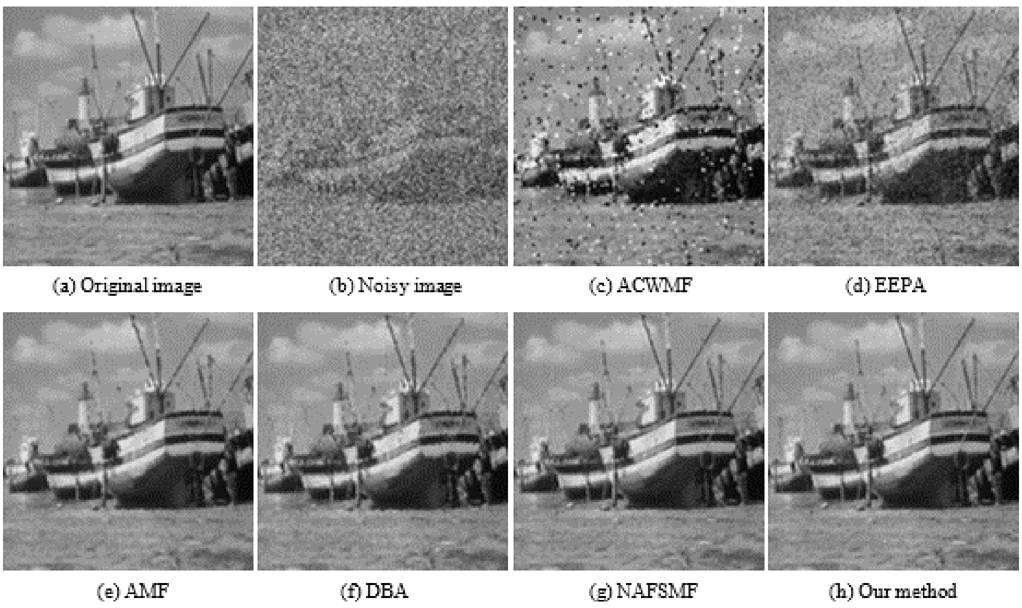

Figure 3: Comparison of visual results for image "Boats".

Meanwhile, the ECMF is employed for the high-densities of noise. The proposed algorithm shows consistent and stable performance across a wide range of noise densities varying from 10%-90%. In contrast to the other existing filters, it produces the best restoration results both visually and quantitatively.

Acknowledgements

The author would like to thank the editors and the anonymous reviewers for their valuable suggestions. This work is supported by the National Natural Science

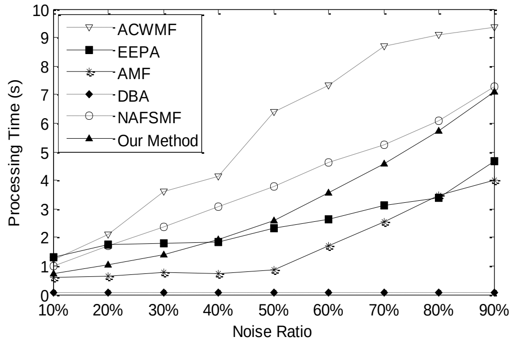

Figure 4: Average processing time (s) versus noise ratios.

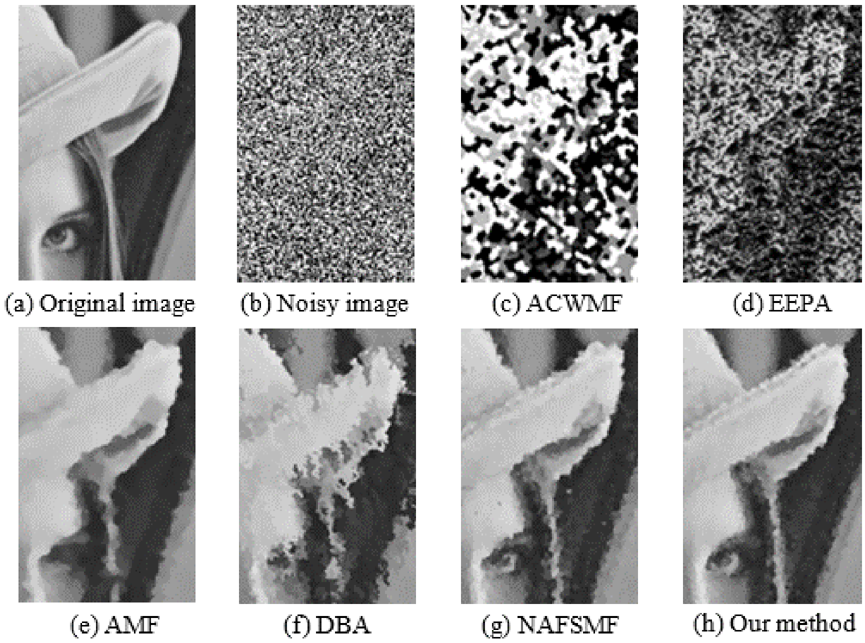

Figure 6: Comparison of visual results for image “Lena”.

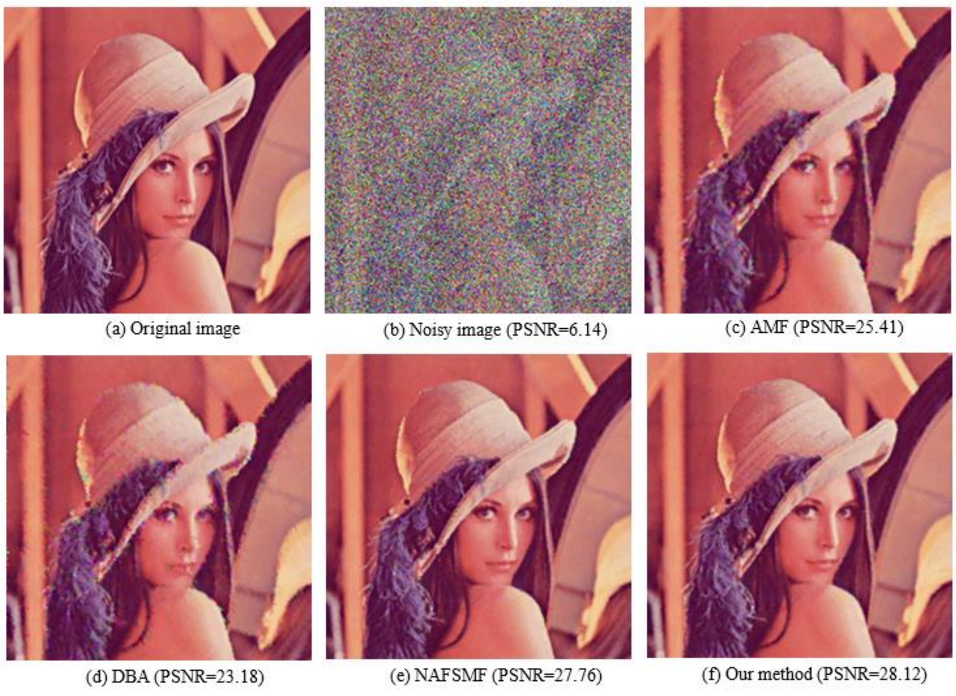

Figure 5: Comparison of visual results for color image “Lena”.

Foundation of China (Grant No. 61471004) and by Doctoral Foundation of Anhui University of Science and Technology (Grant No. ZX942).

References

[1] Huang T. S., Yang G. J., and Tang G. Y. (1979). A fast two-dimensional median filtering algorithm. IEEE Transactions on Acoustics, Speech, and Signal Processing, 27(1), pp. 13-18. https://doi.org/10.1109/tassp.1979.1163188.

[2] Brownrigg D. R. K. (1984). Weighted median filter. Communications of the ACM, 27(8), pp. 807-818. https://doi.org/10.1145/358198.358222.

[3] Ko S. J., and Lee Y. H. (1991). Center weighted median filters and their applications to image enhancement. IEEE Transactions on Circuits and Systems, 38(9), pp. 984-993. https://doi.org/10.1109/31.83870.

[4] Sun T., and Neuvo Y. (1994). Detail-preserving median based filters in image processing. Pattern Recognition Letters, 15(4), pp. 341-347. https://doi.org/10.1016/0167-8655(94)90082-5.

[5] Hwang H., and Haddad R. A. (1995). Adaptive median filters: New algorithms and results. IEEE Transactions on Image Processing, 4(4), pp. 499-502. https://doi.org/10.1109/83.370679.

[6] Chen T., Ma K. K., and Chen L. H. (1999). Tri-state median filters for image denoising. IEEE Transactions on Image Processing, 8(12), pp. 1834-1838. https://doi.org/10.1109/83.806630.

[7] Chen T., and Wu H. R. (2001). Adaptive impulse detection using center-weighted median filter. IEEE Signal Processing Letters, 8(1), pp. 1-3. https://doi.org/10.1109/97.889633.

[8] Srinivasan K. S., and Ebenezer D. (2007). A new fast and efficient decision-based algorithm for removal of high-density impulse noises. IEEE Signal Processing Letters, 14(3), pp. 189-192. https://doi.org/10.1109/ lsp.2006.884018.

[9] Kkishorebabu V., and Varatharajan R. (2020). A decision based unsymmetrical trimmed modified winsorized variants for the removal of high density salt and pepper noise in images and videos. Computer communications, 154, pp. 433-441. https://doi.org/10.1016/j.comcom.2020.02.048.

[10] Thanh D. N. H., Hai N. H., Prasath V. B. S., Hieu L. M., and Tavares J. M. R. S. (2020). A two-stage filter for high density salt and pepper denoising. Multimedia tools and applications, 79(29), pp.21013-21035. https://doi.org/10.1007/s11042-020-08887-6.

[11] Chen P. Y., and Lien C. Y. (2008). An efficient edge-preserving algorithm for removal of salt-and-pepper noise. IEEE Signal Processing Letters, 15, pp. 833-836. https://doi.org/10.1109/lsp.2008.2005047.

[12] Toh K. K. V., and Mat Isa N. A. (2010). Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. IEEE Signal Processing Letters, 17(3), pp. 281-284. https://doi.org/10.1109/ lsp.2009.2038769.

[13] Wang Y., Wang J., Song X., and Han L. (2016). An efficient adaptive fuzzy switching weighted mean filter for salt-and-pepper noise removal. IEEE Signal Processing Letters, 23(11), pp. 1582-1586. https://doi.org/10.1109/ lsp.2016.2607785.

[14] Singh V., Dev R., Dhar N. K., Agrawal P., and Verma N. K. (2018). Adaptive type-2 fuzzy approach for filtering salt and pepper noise in grayscale images, IEEE Transactions Fuzzy system, 26(5), pp.3170-3176. https://doi.org/10.1109/tfuzz.2018.2805289.

[15] Chan R. H., Ho C. W., and Nikolova M. (2005). Salt-and-pepper noise removal by median-type noise detectors and detail preserving regularization. IEEE Transactions on Image Processing, 14(10), pp. 1479-1485. https://doi.org/10.1109/tip.2005.852196.

[16] Thanh D. N. H., Thanh L. T., Hien N. N., and Prasath V. B. S. (2020), Adaptive total variation L1 regularization for salt and pepper image denoising. Optik, 208, pp. 1-10. https://doi.org/10.1016/j.ijleo.2019.163677.

[17] Jayasree P. S., Raj P., Kumar P., Siddavatam R., and Ghrera S. P. (2012). A fast novel algorithm for salt and pepper image noise cancellation using cardinal B-splines. Signal Image Video Process, 5(2), pp. 1-8. https://doi.org/10.1007/s11760-012-0368-3.

[18] Bai T., and Tan J., Hu M., and Wang Y. (2014). A novel algorithm for removal of salt and pepper noise using continued fractions interpolation. Signal Processing, 102, pp. 247-255. https://doi.org/10.1016/j.sigpro.2014.03.023.

[19] Ramadan Z. M. (2012). Efficient restoration method for images corrupted with impulse noise. Circuits, Systems, and Signal Processing, 31, pp. 1397-1406. https://doi.org/10.1007/s00034-011-9380-z.

[20] Enginoglu S., Erkan U., and Memis S. (2019), Pixel similarity-based adaptive riesz mean filter for salt-and-pepper noise removal, Multimedia Tools Applications, 78, 35401-35418. https://doi.org/10.1007/s11042-019-08110-1.

[21] Xu G., and Tan J. (2014). A universal impulse noise filter with an impulse detector and nonlocal means. Circuits, Systems, and Signal Processing, 33, pp. 421-435. https://doi.org/10.1007/s00034-013-9640-1