

An Adaptive Median Filtering of Salt and Pepper Noise based on Local Pixel Distribution

Zhongxin Gao

Department Mathematics and System Science, Shandong University of Science and Technology,
Qingdao, Shandong Province, China

362715609@qq.com

Abstract. A selective adaptive median filter is proposed for the restoration of gray scale images that are highly corrupted by salt and pepper noise. In this paper, impulse noise removal using the standard median filter and its variants are analyzed. After detecting the salt and pepper noise pixels, the self-adaptive median filter is used to find a suitable window containing more non-noise pixels. This proposed algorithm provides much better results than that of the standard median filter, weighed median filter and switching mean median filter. The proposed algorithm is tested against different gray scale images and it gives better Peak Signal Noise Ratio (PSNR), Mean Square Error (MSE) and Structural Similarity Index (SSIM).

Keywords: Impulse noise, noise detection, the self-adaptive median filter, PSNR, MSE, SSIM.

1. Introduction

Image noise removal plays a vital role in image processing as a processing stage [1]. In image processing noise appears from different sources like variation in the detector sensitivity, environmental variation, transmission, and casualization. High levels of noise are always undesirable. Hence noise removal has to be employed before the image could be used for further analysis [2-5].

The available literature shows many methods based on mean and median filters employed for removal of salt and Pepper noise. Some of them are discussed below.

In recent years, many improved algorithms are proposed to overcome the signal pixel and noise pixel. At present, there are many different types of median filtering, such as weighted median filter, switching median filter, adaptive median filter, and adaptive weighted mean filter and so on.

According to the distribution of local pixels, Sue J. (2010) proposed a noise detection algorithm based on Improved Standard Median Filtering (ISMF).

In [6] the authors have used weighted median filter to remove salt and pepper noise without damaging image quality. In [7], median filtering is introduced as boundary discriminant noise detection filter. Here, they achieve a better PSNR value over a wide range of noise density variations between 10% and 70%.

In [8] the authors describe different filters, such as MF, AMF, decision based median filtering, and decision based asymmetric pruning median filters, for removing salt and pepper noise. In [9], adaptive median filtering works well under low noise density. But at higher noise-densities, it leads to large window size which may blur the image.

This algorithm has a good effect on noise suppression, but it often fails when the pixels are destroyed in the window. We propose a selective adaptive median filter to remove salt and pepper noise, which is different from the adaptive median filter [10].

We find that although we can clearly identify the shape of the image, the pixels value varies little in a small range. If we can make full use of the local distribution information to infer the maximum likelihood value of the target pixels, we can improve the PSNR of the restored image [1].

It works as follows: Section 2 describes the proposed adaptive median-based lifting filter. In Section 3, the image denoising algorithm is presented. In Section 4, the significance measurement values of PSNR and SSIM are described to evaluate the quality of the marked image. The efficiency of our proposed method has been shown in Section 5. Finally, Section 6 concludes the work.

2. Median Filter

2.1 Standard Median Filter

A simple median filter [11] uses the median of the window to replace the central pixels considered by the window. If the center pixels are (Pepper) or (salt), it will be replaced by the middle value of the window, which will not be or. The major drawback of standard median filter is that even if the pixels under consideration is uncorrected (other than 0 or 255), it is replaced by the median of the window. This will damage the overall visual quality of the image.

In addition, a simple median filter cannot maintain edges. It works as follows: The window sorted in ascending. Median is the middle value after the sort. Therefore, the undamaged pixels are replaced by the median value of the window.

Sample window Output

$$\begin{bmatrix} 46 & 64 & 82 \\ 255 & (45) & 52 \\ 64 & 64 & 82 \end{bmatrix} \longrightarrow \begin{bmatrix} 46 & 64 & 82 \\ 255 & (82) & 52 \\ 64 & 64 & 82 \end{bmatrix}$$

As shown below, if the considered pixel is destroyed, the impulse noise will be removed in the same way [12].

Sample window Output

$$\begin{bmatrix} 46 & 64 & 82 \\ 255 & (255) & 52 \\ 64 & 64 & 82 \end{bmatrix} \longrightarrow \begin{bmatrix} 46 & 64 & 82 \\ 255 & (82) & 52 \\ 64 & 64 & 82 \end{bmatrix}$$

2.2 Improved Self-adaptive Median Filter

Based on the classical adaptive median filtering algorithm, an improved adaptive median filter (IMF) based on dynamic window size is proposed. It is executed as follows.

(1) The noise pixel $g(x, y)$ are detected by comparing the simple thresholds and averages around the target pixels.

(2) Select the 3×3 window centered at the noise pixels $g(x, y)$. If there exist informative pixels around $g(x, y)$, $g(x, y)$ is replaced by the median value of informative pixels in the window.

(3) If there is not enough information pixels around $g(x, y)$, the window will be expanded to 5×5 (7×7) and repeat the above steps.

Because noisy pixels are replaced by the median values got from informative pixels, the IMF algorithm avoids the spread of noisy signals in the adjacent efficiently during filtering. However, the original pixel distribution is not considered in the recovery process. There is always a significant correlation between adjacent pixel values in natural images.

There are often significant correlations between adjacent pixel values in a nature image. Therefore, noise pixels replaced by adjacent values outside the median value are sometimes more accurate.

3. Proposed Approaches

3.1 Determine the Distance of Effective Pixels

When we analyze the local pixel value distribution, it can be found that a pixel value is often equal or similar to its adjacent pixel values in original digital images.

Table.1 shows the coordinate definition for a window, and Table.2 gives the local pixel value distribution of small image blocks in Lena image

Table.1 A filtering window of size 3×3

| | | |
|--------------|------------|--------------|
| $g(x-1,y-1)$ | $g(x-1,y)$ | $g(x-1,y+1)$ |
| $g(x,y-1)$ | $g(x,y)$ | $g(x,y+1)$ |
| $g(x+1,y-1)$ | $g(x+1,y)$ | $g(x+1,y+1)$ |

Table.2 Local pixel distribution for a small region

| | | | |
|-----|-----|-----|-----|
| 148 | 148 | 148 | 147 |
| 148 | 148 | 149 | 147 |
| 148 | 149 | 149 | 147 |

When focusing on the image block, we find that the values of the adjacent pixel values along the columns, columns or diagonal lines in the local region show certain regularity. We call it a priori knowledge. Furthermore, the following rules are defined for the direction of movement of windows.

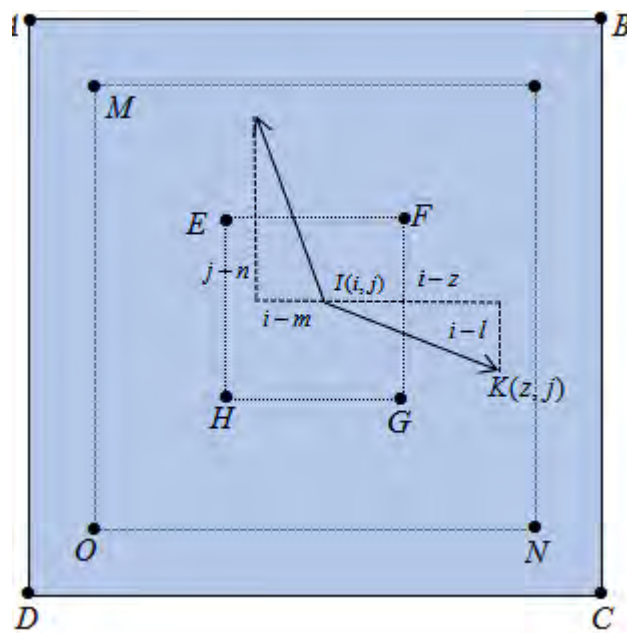


Fig.1 Determine the distance of effective pixels

As shown in Figure.1, the size of the window is determined to be 7×7 directly through the size of the r distance between the center point I and the effective pixel Y coordinates. The results show that the number of sorting decreases and the optimization effect is more obvious when the window is larger. There is a point to be noted here. If the coordinate distance $r = |i - m|$ from I to Y causes the window size to be wrong, the result will be EFCH.

So that Y can't be filtered in the window. Therefore, we should compare the difference between the horizontal and vertical lengths, take a larger $|j - n|$ and make $r = |j - n|$, $Y \in MINO$. Similarly, for H point $r = |i - z|$, this improvement reduces the number of times.

3.2 Noise Detection by Extreme Value Distribution

In order to identify impulse noise, we first define the same matrix as the dimension of the image to be detected [13-18]. A Matrix B is a digital noise image that needs to be detected (where g

represents the location of each point). Let's say, matrix $[a_{ij}]$ is a digital noise image that needs to be detected (where i, j represents the location of each point). T_z Represents a filter window with a size of $(2n+1) \times (2n+1)$ (where n is a positive integer).

T_U Represents a noise detection window. $T[a_{ij}]$ Indicates that a point a_{ij} is centered as a window.

(1) Through noise detection, we can generate a binary symbol image matrix $[K_{ij}]$ which represents the noise distribution in the original image.

If $K_{ij} = 1$ is used in the logo image matrix, the pixel of the original image is impulse noise. On the contrary, if in the logo image matrix $K_{ij} = 0$, the point a_{ij} is the original image pixel.

In the process of noise detection is initialized to all 0 matrices. According to the result impulse noise detection, the element value in the matrix is set to 1 or the original value is 0.

(2) For sorting $N = (2n+1) \times (2n+1)$ points in the noise detection window based on point a_{ij} , we get a vector $B_{ij} = [a_{ij1}, a_{ij2}, a_{ij3}, \dots, a_{ijN}]$.

Where $a_{ij1} < a_{ij2} < \dots < a_{ijN}$. When the sorted point a_{ij} is near the endpoint of vector D_{ij} , point a_{ij} can be a noise point. Finally, the first condition of the noise detection shown in the expression of satisfaction (1) may be the noise point:

$$\begin{cases} G(a_{ij}) \leq m \\ G(a_{ij}) > N - m + 1 \end{cases} \quad (1)$$

In the formula (1), function $G(a_{ij})$ returns the order value of point a_{ij} in the sorted vector D_{ij} . Obviously, there is always a m value that can contain all the noise points in the noise image. On the contrary, however, it is not true. Because of the sorted signal points (which are not contaminated by noise in the image), especially the edge details in the image may also fall near the two ends of the vector D_{ij} . Especially when the PSNR is very high, the false detection rate will be very large.

In order to detect noise points better, we can further examine the specific gray values of point a_{ij} . When the gray value of a point a_{ij} exceeds the range of a parameter T , it may be a noise point. The second conditions of noise detection are as follows:

$$a_{ij} \in [0, T] \text{ or } [255-T, 255] \quad (2)$$

The experimental results show that the gray value of noise-contaminated pixels will change within the range of $[0, 10]$ and $[245, 255]$.

If a certain point in the image satisfies both the formula (1) and the formula (6) at the same time, we can consider it to be a noise point. In this way, the corresponding symbol in the logo image matrix can be 1. Conversely, it is considered a signal point, and the corresponding position in the logo image matrix is labeled as 0. In this way, two input images $[k_{ij}]$ that reflect the noise distribution can be obtained.

$$k_{ij} = \begin{cases} 1, \\ 0, \text{ other} \end{cases} \quad (3)$$

3.3 Filtering Implementation

After detecting the noise pixels, the proposed method focuses on filtering noisy pixels at the boundaries of the image [16]. If a sample (pixel of the image) is noise, it is replaced by the value calculated from other samples in the observation window.

Since there may be more than one pulse in the filter window, the gray value of the noise point should be reconstructed with the nonlinear idea. The method of filtering noise is based on the idea of standard median filtering algorithms. When filtering, first select the filter window T_z . In the noise mark matrix $[k_{ij}]$. For a point, if the point is $k_{ij} = 1$, it means that the point is a noise point. We need to deal with it in two cases.

If there is a signal point in the neighborhood ($n > 0$, n represents the number of signal points), we calculate the median value of the point center filter window.

$$z_{ij} = \begin{cases} med(k(i, j)), n \in odd \\ (k_{med1}(i, j) + k_{med2}(i, j)) / 2, n \in even \end{cases} \quad (4)$$

Among them, n is the number of points marked 0, and k_{ij} is the gray value of points marked 0, represents the gray value of the two points in the middle after sorting.

If there are no image points around the noise points, the points should not be processed. It is directly assigned to the point b_{ij} of the corresponding position of output matrix $[b_{ij}]$.

The noise filtering process can be described as: The filter is first performed with the length of the 3×3 window. The specific operation formula (5) shows that:

$$b_{ij} = \begin{cases} v_{ij}, k_{ij} = 1, n > 0 \\ a_{ij}, else \end{cases} \quad (5)$$

When the noise point a_{ij} is modified by median filtering, it can be regarded as the image point. We need to change the mark from 1 to 0 in the binary sign matrix $[k_{ij}]$ reflecting the noise distribution.

$$k_{ij} = \begin{cases} 0, b_{ij} = v_{ij} \\ 0, \end{cases} \quad (6)$$

When $[b_{ij}]$ is obtained, we examine whether there is a mark in the binary symbol matrix $[k_{ij}]$. If it exists, the image $[b_{ij}]$ is used as the original input image $[a_{ij}] = [b_{ij}]$.

Expand the filter window to 5×5 and repeat the above steps to get $[k_{ij}]$. When there is no point marked in the binary identity matrix $[b_{ij}]$, the loop is stopped. Finally, the image $[b_{ij}]$ is output as the filtering result of the whole algorithm.

4. Significance Measures

4.1 PSNR

We have tested our proposed algorithm for different levels of noise ranging from as low as to as high as 90%. The experimental results have been gauged using the mean square error (MSE) [19] and peak signal-to-noise ratio (PSNR) measures that have been given below [20].

$$MSE = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (A(x, y) - R(x, y))^2 \quad (7)$$

Where A and R are the original and the restored images having a resolution of $m \times n$.

$$PSNR = 10 \log_{10} \left(\frac{\max^2}{MSE} \right) \quad (8)$$

Where \max is the maximum possible pixel value of the image and its value is 255 in the case of a gray scale image.

4.2 SSIM

The SSIM index [21] is a full reference metric, in other words, the measuring of image quality based on an initial uncompressed or distortion free image as reference. SSIM is designed to improve on traditional methods like peak signal to-noise ratio and mean-squared error, which have proved to be inconsistent with human eye perception. SSIM is a new paradigm for quality assessment, based on the hypothesis that the HVS is highly adapted for extracting structural information. The measure of structural similarity compares local patterns of pixel intensities that have been normalized for luminance and contrast. In practice, a single overall index is sufficient enough to evaluate the overall image quality. Hence, a mean SSIM (MSSIM) index is used as the quality measurement metric.

$$SSIM = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_1)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_1)} \quad (9)$$

$$MSSIM = \frac{1}{M} \sum_{m=1}^M SSIM(x_m, y_m) \quad (10)$$

5. Results and Discussion

To prove the advantages of the proposed method, we compare its performance of revising the pepper and salt noise polluted image with several classical median filters which are standard median filter (SMF) [22], Improved Standard Median filter (ISM) [23] and Improved median filter (IMF) [24]. The comparison results got from 512×512 , 8-bits/pixel gray-level Lena image and 512×512 , 8-bits/pixel gray-level retina images with different noise densities (ND) of pepper and salt noise are shown in Fig.2 and Fig.3.

The proposed method is better than the IMF [24] to deal with the edges and details. The performance of different methods is summarized in Fig.4 and Fig.5. Experimental results show that IMF and the proposed method have better performance in filtering the pepper and salt noise than other filters. However, when the noise density increases, the performance of ISM degrades. However,

when the noise density is greater than 0.7, the performance of the proposed method is better than IMF to deal with the edges and details.

In Fig.4, the restoration results have been shown for the noise density 0.3 for corrupted Retina image. Among all these filters, the proposed method gives the best performance in terms of noise suppression and detail preservation.

Using TMF and ISM we are able to remove some amount of noise but a large amount of noisy pixels are present in the output image when noise density is high. Also in IMF some unwanted gray values occur after filtering.

In Fig.2, the restoration results have been shown for the noise density 0.7 for corrupted Lena image.

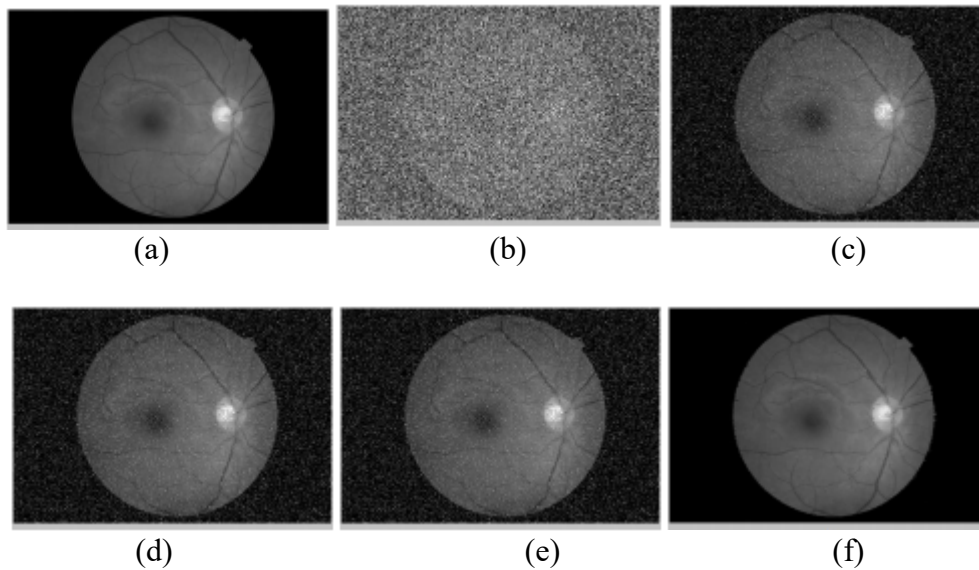


Fig.2 Reconstructed images using the proposed scheme and different

denoising schemes for corrupted Lena image: (a) original noise-free image, (b) 30% noise corrupted, (c) TMF, (d) ISM, (f) proposed scheme.

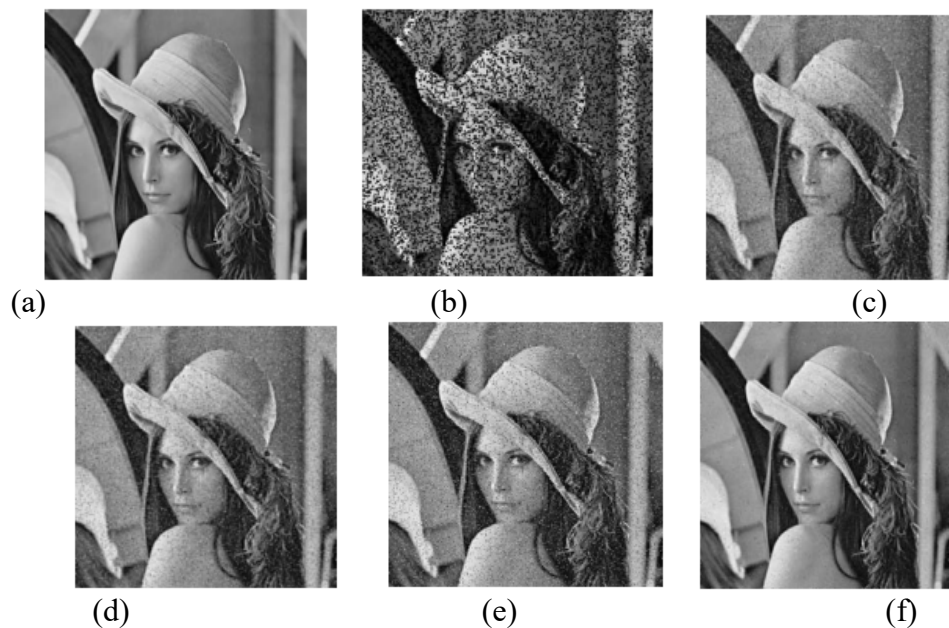


Fig.3 Reconstructed images using the proposed scheme and different denoising schemes for corrupted Lena image: (a) original noise-free image, (b) 70% noise corrupted, (c) TMF, (d) ISM, (e) IMF, (f) proposed scheme.

The proposed method is also compared with other methods in terms of PSNR for high noise levels as shown in table 4. The result shows that the proposed filter performed better than ISM.

It is seen from Tab. 3 that the performance of the proposed method is better than other algorithms at different noise densities from 10% to 90%. Based on SMF, ISM introduced noise detection to discriminate uncorrected pixels from the corrupted pixels. Only the corrupted pixels are filtered. Thus, it provides higher PSNR and lower MSE than IMF.

Table.3 The PSNR values of different denoising methods for Lena image at different noise densities.

| Method | Noise density | | | | | | | | | | | |
|-----------------|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|
| | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 75 | 80 | 85 | 90 | |
| SMF | 30.2 | 29.43 | 28.26 | 27.2 | 25.18 | 23.4 | 20.5 | 18.07 | 16.17 | 14.8 | 12.95 | |
| ISM | 31.2 | 30.52 | 29.58 | 28.46 | 26.85 | 24.15 | 21.2 | 19.31 | 17.57 | 15.93 | 14.3 | |
| IMF | 32.96 | 31.03 | 30.25 | 29.73 | 28.94 | 27.83 | 26.59 | 25.96 | 25.05 | 24.31 | 23.74 | |
| Proposed method | 33.05 | 32.53 | 31.54 | 30.79 | 29.75 | 29.02 | 28.38 | 27.83 | 27.03 | 26.85 | 26.39 | |

Table.4 The MSE values of different denoising methods for Lena image at different noise densities.

| Method | Noise density | | | | | | | | | | | |
|-----------------|---------------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|
| | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 75 | 80 | 85 | 90 | |
| SMF | 9.59 | 11.2 | 12.73 | 13.97 | 16.49 | 21.75 | 29.84 | 37.93 | 45.84 | 56.82 | 68.4 | |
| ISM | 8.61 | 9.79 | 11.07 | 12.73 | 15.95 | 19.74 | 27.38 | 33.54 | 41.3 | 49.73 | 58.72 | |
| IMF | 8.2 | 9.03 | 10.16 | 11.94 | 12.14 | 13.2 | 14.95 | 15.83 | 17.3 | 18.87 | 20.31 | |
| Proposed method | 7.53 | 8.34 | 8.85 | 9.5 | 10.41 | 11.38 | 12.27 | 12.75 | 14.35 | 14.69 | 15.32 | |

The proposed method is also compared with other methods in terms of PSNR for high noise levels as shown in table 4. The result shows that the proposed filter performed better than TMF and ISM.

To demonstrate the performance of the proposed filter, a Lena image with a noise density of 0.7 is adopted. As shown in Table 3, the value of MSE is low, PSNR is high, and SSIM is also very high, which indicates that the performance of the proposed filter is better than that of other filters. In addition, IMF can change the window size of different noise densities to collect enough information pixels to provide better performance.

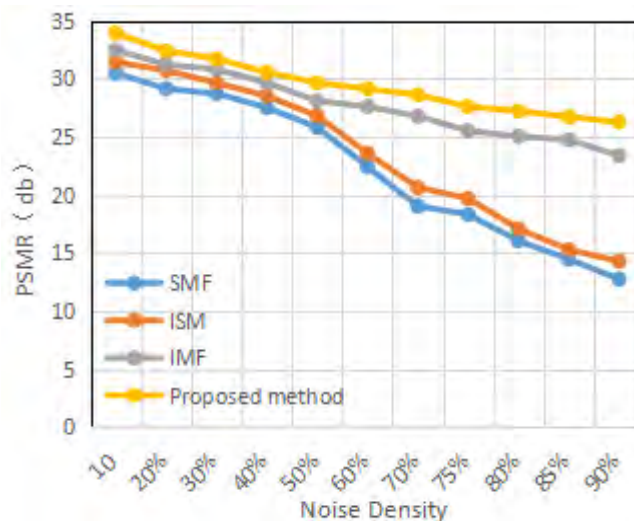


Fig.4 Comparison graph of PSNR at different Noise densities for Lena image

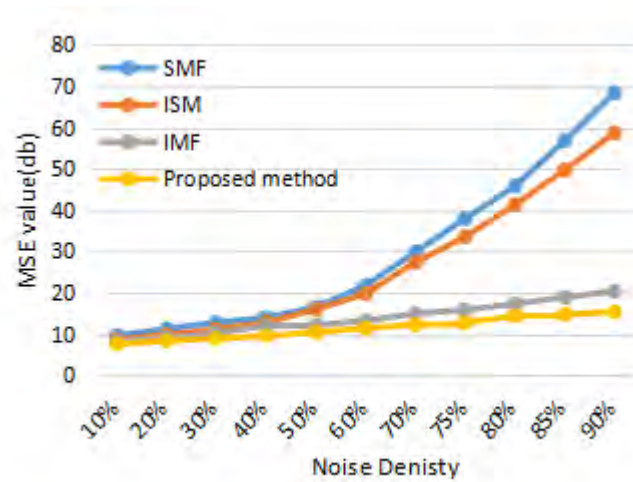


Fig.5 Comparison graph of MSE at different Noise densities for Lena image

Finally, considering the feature of local pixel distribution in the original image, the proposed method extracts noise pixels from the local pixel value, and makes full use of the filtering feature and the correlation between adjacent pixels. Therefore, the experimental results show that the proposed method has the best filtering performance in the four test methods. Figures 4 and 5 shows that the proposed method has better performance and smaller error.

6. Conclusion

A self-adaptive median filter based on local pixel distribution information to remove pepper and salt noise is presented in this paper. The proposed method can not only achieve better image quality, but also have shorter computation time. And our method is simple and easy to be understood. Simulation results reveal that the proposed method provides better performance than the existing method presented for the corrupted by highly noise density in terms of noise suppression and detail preservation. The proposed method gives an efficient filter frame-work, and is suitable for real-time implementation.

Experimental results show that the performance of the proposed method is much superior to that of classical median filter. It performs better than that of the traditional filtering techniques and we hope that our effort will help to improve the future experiments over image processing and performance analysis. In future we will try to explore the effect of other filtering techniques over noisy image and upgrade them according to achieve the better performance.

References

- [1]. Sadhar, S. Ibrahim, and A. N. Rajagopalan. "Image estimation in film-grain noise." *IEEE Signal Processing Letters* 12.3 (2005): 238-241.
- [2]. Leavline, E. Jebamalar, and S. Sutha. "Gaussian noise removal in gray scale images using fast Multiscale Directional Filter Banks." *Recent Trends in Information Technology (ICRTIT)*, 2011 International Conference on. IEEE, 2011.
- [3]. Chahal, Gurpreet, and Harminder Singh. "Robust statistics based filter to remove salt and pepper noise in digital images." *International Journal of Information Technology and Knowledge Management* 2.2 (2010): 601-604.
- [4]. Leavline, E. Jebamalar, S. Sutha, and D. Asir Antony Gnana Singh. "Wavelet domain shrinkage methods for noise removal in images: A compendium." *International Journal of Computer Applications* 33.10 (2011): 28-32.

- [5]. Luo, Wenbin. "Efficient removal of impulse noise from digital images." *IEEE Transactions on Consumer Electronics* 52.2 (2006): 523-527.
- [6]. Ng, Pei-Eng, and Kai-Kuang Ma. "A switching median filter with boundary discriminative noise detection for extremely corrupted images." *IEEE Transactions on image processing* 15.6 (2006): 1506-1516.
- [7]. Esakkirajan, S., et al. "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter." *IEEE Signal processing letters* 18.5 (2011): 287-290.
- [8]. Hwang, Humor, and Richard A. Haddad. "Adaptive median filters: new algorithms and results." *IEEE Transactions on image processing* 4.4 (1995): 499-502.
- [9]. Morillas, Samuel, et al. "New adaptive vector filter using fuzzy metrics." *Journal of Electronic Imaging* 16.3 (2007): 033007.
- [10]. Hwang, Humor, and Richard A. Haddad. "Adaptive median filters: new algorithms and results." *IEEE Transactions on Image Processing* 4.4 (1995): 499-502.
- [11]. Shukla, H. S., Narendra Kumar, and R. P. Tripathi. "Median Filter based Wavelet Transform for Multilevel Noise." *International Journal of Computer Applications* 107.14 (2014): 11-14.
- [12]. R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Pearson Prentice Hall Publication, (2008).
- [13]. Yuan, Shi-Qiang, Yong-Hong Tan, and Hua-Li Sun. "Impulse noise removal by the difference-type noise detector and the cost function-type filter." *Signal Processing* 87.10 (2007): 2417-2430.
- [14]. Chan, Raymond H., Chungwa Ho, and Mila Nikolova. "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization." *IEEE Transactions on Image Processing* 14.10 (2005): 1479-1485.
- [15]. Zhang, Xuming, and Youlun Xiong. "Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter." *IEEE Signal processing letters* 16.4 (2009): 295-298.
- [16]. Schulte, Stefan, et al. "A fuzzy impulse noise detection and reduction method." *IEEE Transactions on Image Processing* 15.5 (2006): 1153-1162.
- [17]. Camarena, Joan-Gerard, et al. "Fast detection and removal of impulsive noise using peer groups and fuzzy metrics." *Journal of Visual Communication and Image Representation* 19.1 (2008): 20-29.
- [18]. Huang, Thomas S., George J. Yang, and Gregory Y. Tang. "A fast two-dimensional median filtering algorithm." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.1 (1979): 13-18.
- [19]. Guo, Dongning, et al. "Estimation in Gaussian Noise: Properties of the Minimum Mean-Square Error." *IEEE Transactions on Information Theory* 57.4 (2011): 2371-2385.
- [20]. Aizenberg, Igor, Constantine Butakoff, and Dmitriy Paliy. "Impulsive noise removal using threshold Boolean filtering based on the impulse detecting functions." *IEEE Signal Processing Letters* 12.1 (2005): 63-66.
- [21]. Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." *IEEE Transactions on Image Processing* 13.4 (2004): 600-612.
- [22]. Veerakumar, T., S. Esakkirajan, and Ila Vennila. "Salt and pepper noise removal in video using adaptive decision based median filter." *multimedia signal processing* (2011): 87-90.

- [23]. Zhu, Youlian, and Cheng Huang. "An Improved Median Filtering Algorithm for Image Noise Reduction." *Physics Procedia* (2012): 609-616.
- [24]. Zhang, Shuqun, and Mohammad A. Karim. "A new impulse detector for switching median filters." *IEEE Signal processing letters* 9.11 (2002): 360-363.