Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 1388 – 1403
© TÜBİTAK
doi:10.55730/1300-0632.3855

A new effective denoising filter for high density impulse noise reduction

Iman ELAWADY[1,2], **Caner ÖZCAN**[2,*]
[1]Department of Electrical Engineering, Ecole Nationale Polytechnique d'Oran Maurice Audin, Oran, Algeria
[2]Department of Software Engineering, Faculty of Engineering, Karabuk University, Karabuk, Turkey

Received: 05.08.2021 • **Accepted/Published Online:** 17.03.2022 • **Final Version:** 31.05.2022

Abstract: Today, thanks to the rapid development of technology, the importance of digital images is increasing. However, sensor errors that may occur during the acquisition, interruptions in the transmission of images and errors in storage cause noise that degrades data quality. Salt and pepper noise, a common impulse noise, is one of the most well-known types of noise in digital images. This noise negatively affects the detailed analysis of the image. It is very important that pixels affected by noise are restored without loss of image fine details, especially at high level of noise density. Although many filtering algorithms have been proposed to remove noise, the enhancement of images with high noise levels is still complex, not efficient or requires very long runtime. In this paper, we propose an effective denoising filter that can restore the image effectively in terms of quality and speed with less complexity for high density noise level. In the experimental studies, we compare the denoising results of the proposed method with other state-of-the-art methods and the proposed algorithm is quantitatively and visually comparable to these algorithms when the noise intensity is up to 90%.

Key words: Noise reduction, image filtering, salt and pepper noise, median filter, impulse noise

1. Introduction

With the increasing speed and data storage capacity of computers and other signal processors, the interest in image processing research is increasing day by day. However, digital images are exposed to impulse noise, which affects the detail properties of the image and visually reduces its quality during acquisition through sensors, wired or wireless transmission, and storage. Salt and pepper noise (SPN), a common impulse noise, is the random presence of pixels with an extreme value (minimum or maximum) in an image [1]. Even at low noise density, SPN can significantly degrade the visual quality of the image. It is not only degrading the visual quality, but also the performance of subsequent image processing such as edge detection [2], classification [3], segmentation [4], and recognition [5]. Repairing damaged pixels before analysis of the image is an essential research task of digital image processing. Hence, noise removing is a preliminary step for enhancing degraded images, and its main purpose is to preserve detailed features such as edges, points, textures and other properties on images while reducing noise.

In cases where the noise type is nonadditive, linear filters do not provide sufficient performance, while nonlinear methods are proven to be more successful. The standard median filter (MF) [6], which forms the basis of nonlinear methods, is widely used due to its performance in reducing low-density noise, high computational efficiency, and simplicity. The noisy intensity value of the center pixel of the window is removed by replacing

[*]Correspondence: canerozcan@karabuk.edu.tr

the calculated median intensity value of the pixels inside the window. While it can effectively reduce SPN at low densities of noise, it performs poorly at higher densities of noise. Pixels without noise can also be considered noise, as MF applies the same treatment to all pixels in the image. In addition, larger sized windows should be used for images with high-density noise. Such situations cause the MF filter to eliminate desirable features such as edges and textures.

To eliminate these drawbacks, improved versions of the median filter have been proposed, such as adaptive median filter (AMF) [7], weighted median filter (WMF) [8], center-weighted mean filter (CWMF) [9], and decision-based filtering algorithm [10] are proposed. In these methods, variable window size and different weight values for each neighborhood are used to reconstruct the gray level of the middle pixel in the window of all pixels in an image. Therefore, these filters can achieve a better restoration result than SMF. Another variant of the MF filter [11] is also effective at low intensity noise. While applying the filter, firstly, the pixels affected by the noise are determined with the adaptive median filter, and then a special regularization process is applied to these pixels. Although these variations and improvements offered better performance than MF at low intensity noises, they did not provide adequate performance for high intensity noises.

Switching filters, which is one of the important uses of the median filter, divides the filtering process into two steps as noise detection and filtering. Since these methods use the thresholding, only the value of the noisy pixel changes, while no change is made in the nonnoisy pixels. If the difference is greater than a certain threshold, the central pixel will be considered a noisy pixel. In this way, the method ensures that the image details are protected successfully. The basis of these methods is switching median filter (SMF) [12]. New versions of this filter were presented with the improved studies carried out later such as progressive switching median filter [13], modified switching median filter (MSMF) [14], adaptive switching median filter (ASMF) [15], adaptive switching weighted median filter (ASWMF) [16] and switching-based adaptive weighted mean filter (SAWMF) [17]. In these methods, besides using a direction difference-dependent noise detector to locate noisy pixels, denoised pixel value is obtained by replacing each corrupted pixel with the uncorrupted neighbor weighted average in the filter window. Although significant efficiencies regarding noise detection, image noise removal, and computational efficiency are achieved, the selection of thresholds is very random depending on the image and noise density, and when the noise density is high, image details and edges cannot be fully preserved.

The state-of-the-art MF filters have been also provided for obtaining better denoising performance. Among many other SPN reducing methods, directional weighted median filter [18], the improved median filter [19], adaptive weighted mean filter (AWMF) [20], improved adaptive median filters [21] are implemented. Later, the modified directional weighted median filter [22] is proposed, which replaces only the distorted pixels in the image and replaces the neighborhood information of each pixel with the median of the pixel values in all four main directions. A modified decision-based unsymmetric trimmed MF [23] has been proposed to remove the salt and pepper noise, which detects pixels degraded by noise due to excessive gray level. It is provided adaptively to replace the noisy pixel with the mean or median value according to the number of excessive gray levels in the filtering window. Although modified decision-based unsymmetric trimmed MF provides improved image quality at low noise, it causes greater error at higher noise density.

Lately, a new filtering algorithm BPDF [24] that takes pixel density into account has been proposed for the degraded image that can only be used efficiently for lower-density and medium-density SPN. In this filter, high level of noise causes a raindrop effect and distorts the image properties. Besides these, algorithms [25, 26] based on fuzzy sets improved the detection accuracy by setting fuzzy threshold values. The noise adaptive fuzzy SMF [27] provides a fuzzy function which decides the positions and the values to fill the noisy pixels. A new adaptive Type-2 fuzzy filter (FDS) is defined by categorizing pixels based on their primary membership function values in the given window [28]. The fuzzy directional median filter [26] performs the detection of smooth and nonsmooth regions as a preliminary step and tries to detect noisy pixels in these regions. It works by using the smallest standard deviation value for the detection of regions. However, this filter is not sufficient to eliminate the noise above the medium level.

In addition to the filters mentioned above, filters that are useful at high noise intensities and in situations with many noisy pixels in the window are also proposed. Adaptive switching nonlocal filter (ASNLF) [29] is proposed by addressing the limitations of adaptive switching and nonlocal-based algorithms. A simple and efficient filter [30], which is a union of the median and the mean filter, provided a high rate of noise removal success at high-density noises but unsuccessful at lower levels. The probabilistic decision-based filter [31] is presented to decrease noise applying patch else trimmed median by combining two algorithms. In another study, a different applied MF [32] is proposed considering the value of neighboring pixels and the adaptive window. An adaptive decision-based kriging interpolation technique (ADKIF) is proposed [33] by replacing a value that is interpolated using the weights calculated using semivariance between the noisy and the nonnoisy pixels in a defined neighborhood.

Adaptive Riesz mean filter (ARmF) [34], adaptive Cesáro mean filter (ACmF) [35], improved adaptive weighted mean filter (IAWMF) [36] and different adaptive modified Riesz mean filter (DAMRmF) [37] which employ special weighted means for salt and pepper image are proposed. Besides these filters, adaptive frequency median filter (AFMF) [38] is applied using frequency median to restore gray values of the corrupted pixels. These proposed filters have successfully reduced SPN at different intensities. In recent years, regularization-based approaches [39, 40] have been also proposed. The results obtained in TVL1 [40] show that the proposed approach significantly reduces noise in images with intense noise, while successfully preserving the edges and fine properties of the original image. Few studies are presented that address the issue of noise reduction related to SPN using a deep learning approach [41, 42]. Thanks to the preprocessing step, the presence of high-density noise is reduced, resulting in high-quality noise-free images.

Based on the above studies, noise removal at high densities while preserving image detail characteristics is still a very challenging task. Existing restoration filters are not capable of producing higher quality denoised images from distorted images at higher density of noise and large window size, as observed in recent studies [43, 44]. Due to these drawbacks, we try to propose an effective denoising filter for high density SPN reduction with high performance. As the details are given in the second section, our novelty in this new method is to generate a new pixel value over the mean and median information by using a small 3x3 window between the uncorrupted neighbor pixels of the noisy pixel. Keeping the window size especially small brings along significant gains in processing times. When we compare the proposed method with other algorithms in the literature, it stands out with its lower complexity and running time. The proposed algorithm has the characteristics of being easily applicable in different noise removal applications with performance requirements, as it has an uncomplicated flow and is simple in terms of coding. The results of extensive experimental studies show that the reconstruction quality of our algorithm is comparable to other state-of-the-art algorithms, both visually and based on PSNR and SSIM metrics, even when the noise density is up to 90%. In addition, it is clearly seen that the proposed method makes a significant contribution to the literature in terms of processing time.

The rest of this paper is organized as follows. The second part introduces the salt and pepper noise with some statistical parameters and provides an overview of the proposed method. The third part describes the experimental results in detail, and performance of the proposed algorithm is discussed. In the last part, we shared the conclusions to summarize the study.

2. Proposed approach

In this section, we first explained the statistical properties of salt and pepper noise in images, and then explained the details of our proposed method.

2.1. Characteristics of noise

Many degradations can occur during the capture, transmission, or compression of a picture, generating parasitic information that might deteriorate the image. We term this information "noise". Images are employed in a wide variety of areas, including medical, manufacturing, photography, and so on. It is useful to understand the many types and sources of noise in order to avoid or remove it. There are three sorts of noise in general.

- Additive noise: g(x, y) = f(x, y) + b(x, y) ;
- Multiplicative noise: g(x, y) = f(x, y) x b(x, y);
- Convolutional noise: g(x, y) = f(x, y) * b(x, y).

where: g (x, y): the noisy image, f (x, y): the original image, b (x, y): the noise.

The additive noise is the most common noise in image processing. In salt and pepper noise, random bright (with maximum value) and dark (with minimum value) pixels in the image data occur where sharp and sudden distortions (from image sensor and circuitry of a scanner or digital camera etc.) are seen in the image signal. Generally, we characterize the noise according to the probability density function (PDF) which is given as follows [1]:

$$P(z) = \begin{cases} P_a & if \quad z = a \\ P_b & if \quad z = b \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where z is gray level and if b > a then b is light dot (255) and a is dark dot (0). If either P_a or P_b is zero, it is unipolar impulse noise, otherwise it is bipolar impulse noise. The noise properties are given in Figure 1.

In this model, if the value of a is smaller than the value of b on the z axis, the gray level a will appear as a dark spot in the image. Again in this case, the gray level b will appear as a lighter point. If any of the densities a and b are zero, the noise is unipolar. In the case where the probabilities are not equal to zero, but equal, the noise type will be randomly scattered in the image. In cases where the noise is nonunipolar, that is, bipolar, impulse noise is considered salt and pepper noise.

2.2. Proposed denoising algorithm

In this section, we provide some basic concepts that we used while describing our algorithm. Then, the pseudocode and flow diagram of our algorithm for the proposed algorithm is given and it is explained how noise removal is achieved. Our algorithm uses fixed window size. The center pixel of the window is considered noisy if its value is equal to the salt or the pepper; otherwise, it is a noise-free pixel. The proposed algorithm generates a new pixel value using the mean and median information between the uncorrupted neighbor pixels of the noisy pixel. In this study, salt and pepper noise, which is a variant of impulse noise assumed to be randomly distributed. Salt and pepper effect are seen on an 8-bit image with equal probability.

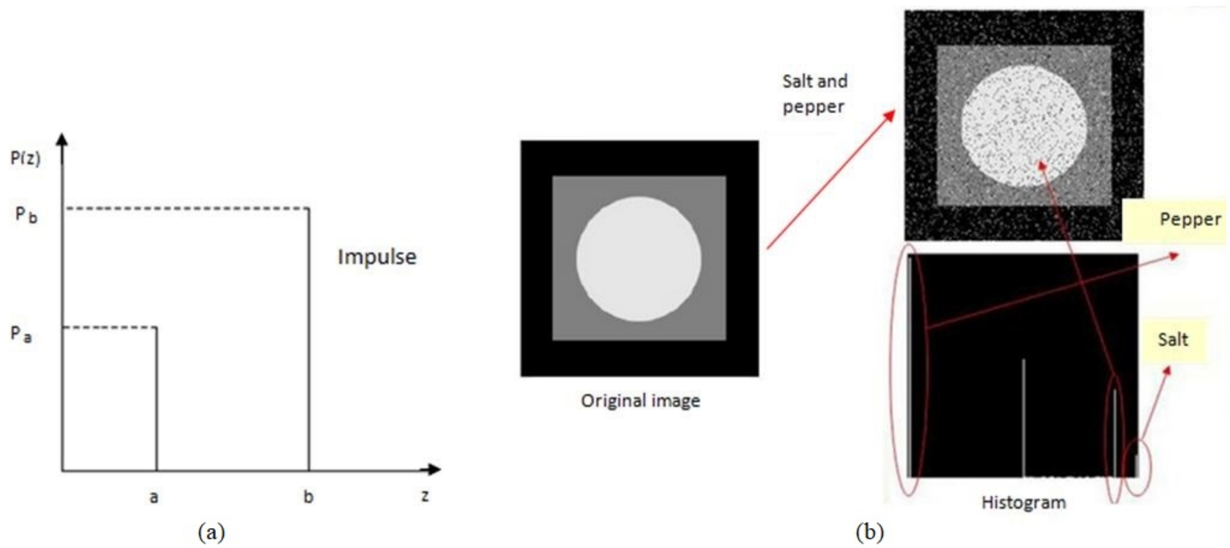


Figure 1. The properties of impulse noise: (a) power density function of salt and pepper noise, (b) example of salt and pepper noise with histogram

Definition 2.1 Assume $A := [a_{ij}]_{mxn}$ is an image and a_{ij} is an unsigned integer number between 0 and 255. Here, m is the number of rows and n is the number of columns in image A.

$$\forall\ i = \ [1, 2, ...m]\,,\ j = [1, 2, ...n] \Rightarrow a_{ij} \in [0, 255] \tag{2}$$

Definition 2.2 a_{ij} is called as noisy pixel of A if $a_{ij} = 0$ or $a_{ij} = 255$; otherwise, a_{ij} is called as nonnoisy pixel of A.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & \cdots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & \cdots & & & \cdots & a_{2,n-1} & a_{2,n} \\ \vdots & \ddots & & & & & \vdots \\ & & & & & & \\ \vdots & & & & & \ddots & \vdots \\ a_{m-1,1} & \cdots & & & & \cdots & a_{m-1,n} \\ a_{m,1} & a_{m,2} & \cdots & & \cdots & a_{m,n-1} & a_{m,n} \end{bmatrix} \tag{3}$$

Example 2.1 Let A matrix as

$$\begin{bmatrix} 49 & 51 & 50 & 49 & 49 & 49 & 48 & 49 \\ 52 & 51 & 49 & 47 & 49 & 49 & 48 & 48 \\ 52 & 51 & 48 & 48 & 51 & 49 & 49 & 49 \\ 51 & 50 & 49 & 49 & 51 & 50 & 49 & 49 \\ 52 & 51 & 50 & 50 & 49 & 49 & 49 & 49 \\ 52 & 50 & 52 & 51 & 51 & 50 & 50 & 48 \\ 50 & 50 & 51 & 51 & 52 & 50 & 48 & 47 \\ 50 & 49 & 50 & 50 & 49 & 49 & 49 & 47 \end{bmatrix}$$

Definition 2.3 Let window w sized 3×3 denoted by $w(i-1:i+1, j-1:j+1)$; w can be shifted horizontally by $j+1$ and vertically by $i+1$.

$$w = \begin{bmatrix} A(i-1,j-1) & A(i-1,j) & A(i-1,j+1) \\ A(i,j-1) & A(i,j) & A(i,j+1) \\ A(i+1,j-1) & A(i+1,j) & A(i+1,j+1) \end{bmatrix} \tag{4}$$

Example 2.2 Example of w is shifting horizontally

$$\begin{bmatrix} 49 & 51 & 50 & 49 & 49 & 49 & 48 & 49 \\ 52 & 51 & 49 & 47 & 49 & 49 & 48 & 48 \\ 52 & 51 & \overrightarrow{\overline{48 \quad 48 \quad 51}49} & 49 & 49 \\ 51 & 50 & \overrightarrow{\overline{49 \quad 49 \quad 51}50} & 49 & 49 \\ 52 & 51 & \overrightarrow{\overline{50 \quad 50 \quad 49}49} & 49 & 49 \\ 52 & 50 & 52 & 51 & 51 & 50 & 50 & 48 \\ 50 & 50 & 51 & 51 & 52 & 50 & 48 & 47 \\ 50 & 49 & 50 & 50 & 49 & 49 & 49 & 47 \end{bmatrix}$$

In the proposed algorithm, we use two simple statistical parameters for removing the noise: mean value and median value. Let A be an Im and window size is $(2k+1) \times (2k+1)$, the function defined for the mean value is as follows:

$$f_1\left(x,y\right) = \frac{1}{(2k+1)^2} \sum_{i=-k}^{k} \sum_{j=-k}^{k} Im\left(x-i,y-j\right) \tag{5}$$

The coefficient in front of the equation attributes uniform weight to each pixel. Thanks to the sum operations, the loop operates all pixels in the neighborhood around image pixel $Im(i,j)$. We denoted $(2k+1)$ as the size of selected window and $(2k+1)$ is odd and symmetrical. In our case where $k=1$,

$$f_1\left(x,y\right) = \frac{1}{9}[Im\left(x-1,y-1\right) + Im\left(x-1,y\right) + Im\left(x-1,y+1\right) + Im\left(x,y-1\right) + \\ Im\left(x,y\right) + Im\left(x,y+1\right) + Im\left(x+1,y-1\right) + Im\left(x+1,y\right) + Im\left(x+1,y+1\right)]. \tag{6}$$

For the median value, we identify the function $f_2(x,y)$ described as:

$$f_2\left(x,y\right) = median\left[Im\left(x-i,y-j\right)\right] \qquad i,j = [-k,k] \tag{7}$$

The median represents middle of a sorted list of numbers, and it is calculated by arranging to the data values from the lowest to the highest value; in this case the median will be the data value in the middle of the set. The flowchart of our algorithm is given in Figure 2 and its main steps are as follows:

In our proposed algorithm, we first apply the detection step to identify the possible noise in the image and then the correction step to remove this noise. As we mentioned in the introduction part, we use the window size as 3×3 in our method. Especially keeping the window size small in this way makes a significant contribution to our algorithm in the processing time. Different sizes of the window can be also used, for example, 5×5, 7×7, etc. However, a smaller sized window creates a lower blur effect, while a larger sized window produces a high blur effect.

In the first three steps of the proposed algorithm, reading the noisy image, converting the image to fractional for fractional operations, and performing the initialization of the output image as the input image,

Algorithm 1 Proposed Algorithm Steps

```
Step 1. Read the noisy image $I_{in}$
Step 2. Convert $I_{in}$ from uint8 to double
Step 3. Initialize $I_{out}=I_{in}$ and set the size of w to $3\times3$
Step 4. Repeat
    Select window w((i-1):(i+1),(j-1):(j+1))
    If (w(i,j)==0 || w(i,j)==255)
        If (all other pixels in w == (0||255))
            $I_{out}$(i,j)= median(w((i-1):(i+1),(j-1):(j+1)))
        Else
            $I_{out}$(i,j)= mean(rest of pixels in w which $\neq$ (0||255))
Until Each pixel in $I_{in}$ visited
Step 5. Convert $I_{out}$ from double to uint8
```

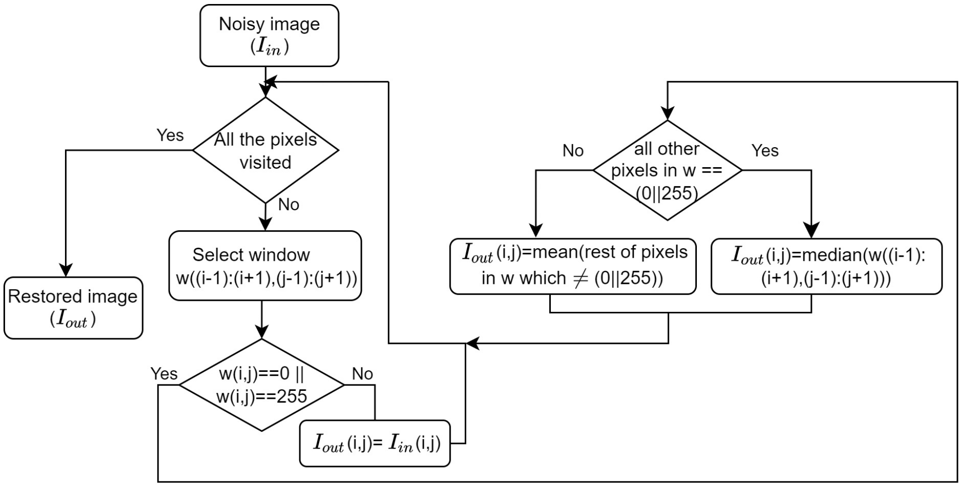

Noisy image (I_{in})
All the pixels visited
Yes
No
Restored image (I_{out})
Select window w((i-1):(i+1),(j-1):(j+1))
w(i,j)==0 || w(i,j)==255
Yes
No
I_{out}(i,j)= I_{in}(i,j)
all other pixels in w == (0||255)
No
Yes
I_{out}(i,j)=mean(rest of pixels in w which ≠ (0||255))
I_{out}(i,j)=median(w((i-1):(i+1),(j-1):(j+1)))

Figure 2. Flowchart of the proposed algorithm.

respectively. In the fourth step of the algorithm, a loop is created that will visit all pixels in the image. In this loop, a window is first created for the 3×3 neighborhood of the relevant central pixel. Then it is checked whether the center pixel has salt (equal to 255) or pepper noise (equal to 0). If the center pixel value is not salt or pepper, the value of the new pixel in the output image is equal to the value of this center. Otherwise, all pixels in the existing window are controlled. If all other pixels in the window are salt or pepper, the new pixel value is found by calculating the median of all pixels in this window. In the else case, the new pixel value is found by calculating the average of all pixels in the window that are not salt or pepper. After all the pixels in the image are visited, the creation of the new image is completed, and in the last step, the image is converted from double to integer, and the processes are completed.

3. Experimental studies

There are many evaluation parameters that can be used to certify the obtained results. In our simulation results, first peak signal-to-noise ratio (PSNR) and second structural similarity index measure (SSIM) is obtained. PSNR acts as an image quality estimator between two images and calculates the peak signal to noise ratio. It is most commonly used to estimate the efficiency of image quality between the distorted and the original image. A high PSNR value indicates a low degree of image degradation [45]. For the mathematical notations of PSNR and SSIM, let $X = [x_{ij}]$ be the original image and $Y = [y_{ij}]$ be the denoised image. We can define PSNR as:

$$PSNR= 10.log_{10}\left(\frac{255^2}{MSE(X,Y)}\right) \tag{8}$$

MSE is the cumulative squared error which measures the average of the squares of the errors between the original and the denoised image. The lower the MSE obtained, the higher the accuracy.

$$MSE(X,Y)=\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}-y_{ij})^2 \tag{9}$$

The SSIM calculates the degradation of structural information between two images on grayscale image. If the result tends towards 1, this indicates that the image has a good quality. The overall index is a multiplicative combination of the three terms [45].

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{10}$$

μ_x is the mean of x, μ_y is the mean of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, σ_{xy} is the covariance of x and y, C_1=$(k_1L)^2$ with k_1= 0.01, C_2= $(k_2L)^2$ with k_2= 0.03 and L: is the pixel-values between 0-255.

For the experimental studies, we firstly used standard test images such as Lena, Peppers, Plane, Boat, and Goldhill with sized 512×512, in the gray level (8-bit encoding), window sized 3×3, and with different noise densities. Various sizes of the window such as 5 × 5, 7 × 7 can be used, but a smaller window is preferred because it produces less blur effect. Using these images, the performance of the proposed method was evaluated and a comparison was made with other existing methods. The numerical results are represented as shown below using SSIM and PSNR as given in equations 8 and 10. Our results were compared with standard median filter (MF) [6], decision-based algorithm (DBA) [10], progressive switching median filter (PSMF) [13], modified decision-based unsymmetric trimmed median filter (MDBUTMF) [23], different applied median filter (DAMF) [32] and noise adaptive fuzzy switching median filter (NAFSMF) [25] as shown below. The obtained results represent the mean value of ten iterations with random noise. According to Tables 1–3, the proposed algorithm produced better results in Lena, Peppers, and Plane images for each metric.

Figures 3–5 show graphical illustrations of the proposed method with different noise densities for each test images. Noise densities were chosen as 50%, 70%, and 90%, respectively, for the generation of visual results.

According to the obtained results, the proposed method gives effective results for the possible noise intensities for the different input images used. It has been observed that the qualitative performance of the proposed algorithm on the test images is particularly good at the intense noise level.

In addition, we performed studies on 20 traditional images (Baboon, Barbara, Blonde Woman, Boat, Bridge, Cameraman, Dark-Haired Woman, Einstein, Elaine, Flintstones, Flower, Hill, House, Lake, Lena, Living Room, Parrot, Peppers, Pirate, and Plane) and 40 test images in TESTIMAGES Database [46] to show the performance of our algorithm in detail. In Tables 4 and 5, PSNR (dB) and SSIM results are given for different levels of noise removal using state-of-the-art methods in the literature.

Table 1. Simulation results of different filters based on PSNR (dB)/SSIM for Lena image.

Noise (%)	MF	DBA	PSMF	MDBUTMF	NAFSMF	DAMF	Proposed
10	33.02/0.98	30.22/0.97	36.4/0.98	37.91/0.95	38.93/0.98	**42.98/0.99**	42.45/0.99
20	28.74/0.93	28.39/0.94	32.9/0.96	34.78/0.86	35.65/0.96	39.24/0.97	**39.57/0.99**
30	23.54/0.78	25.52/0.89	30.15/0.91	32.29/0.84	33.66/0.94	36.68/0.96	**37.08/0.99**
40	18.94/0.52	22.49/0.83	28.49/0.78	30.32/0.84	32.39/0.92	34.95/0.95	**35.39/0.98**
50	15.23/0.31	19.31/0.75	26.41/0.56	28.18/0.82	30.96/0.90	33.24/0.93	**33.81/0.98**
60	12.35/0.18	12.10/0.66	24.83/0.11	26.43/0.81	29.91/0.88	31.76/0.90	**32.20/0.97**
70	10.01/0.10	9.84/0.56	22.64/0.05	24.30/0.75	28.72/0.84	30.32/0.87	**30.60/0.95**
80	8.14/0.05	8.02/0.44	20.32/0.02	21.70/0.73	27.14/0.80	28.47/0.83	**28.78/0.93**
90	6.64/0.02	6.57/0.34	17.14/0.01	18.40/0.33	23.62/0.68	26.04/0.76	**26.36/0.88**

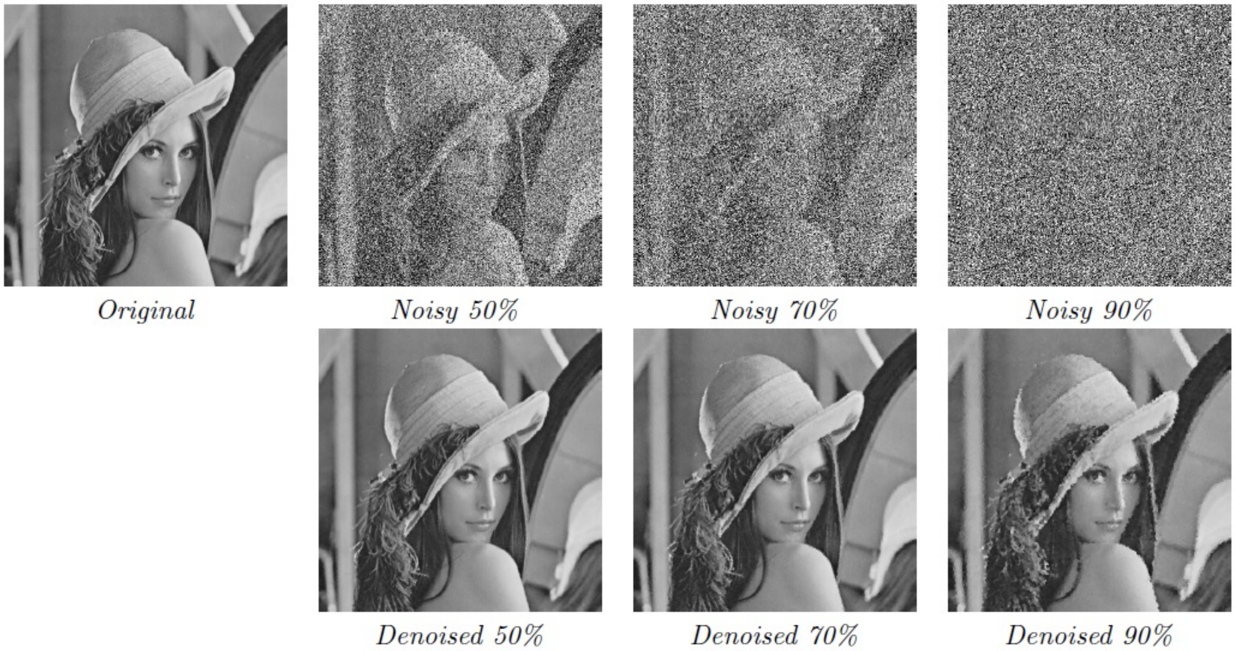
Table 2. Simulation results of different filters based on PSNR (dB)/SSIM for Peppers image.

Noise (%)	MF	DBA	PSMF	MDBUTMF	NAFSMF	DAMF	Proposed
10	32.82/0.97	36.75/0.95	36.18/0.96	35.88/0.94	39.48/0.97	41.41/0.98	**43.67/0.99**
20	28.71/0.93	32.87/0.91	31.32/0.92	33.30/0.89	36.32/0.95	37.99/0.96	**40.39/0.99**
30	23.46/0.78	29.36/0.85	27.71/0.85	31.24/0.82	34.36/0.93	35.56/0.93	**38.22/0.99**
40	18.84/0.53	26.89/0.78	24.57/0.74	31.01/0.81	32.81/0.90	33.86/0.91	**36.44/0.99**
50	15.13/0.31	24.30/0.70	15.06/0.24	30.56/0.79	31.69/0.88	32.53/0.88	**34.81/0.98**
60	12.23/0.18	21.48/0.61	12.17/0.11	29.25/0.76	30.41/0.85	31.19/0.85	**33.14/0.98**
70	9.89/0.10	18.51/0.50	9.78/0.05	28.71/0.73	29.00/0.81	29.80/0.81	**31.47/0.97**
80	8.01/0.05	15.18/0.38	7.96/0.02	24.29/0.70	27.43/0.76	28.29/0.77	**29.54/0.95**
90	6.49/0.02	12.18/0.28	6.49/0.01	15.61/0.34	23.69/0.65	26.04/0.70	**26.81/0.90**

Table 3. Simulation results of different filters based on PSNR (dB)/SSIM for Plane image.

Noise (%)	MF	DBA	PSMF	MDBUTMF	NAFSMF	DAMF	Proposed
10	31.91/0.98	36.25/0.98	33.79/0.96	34.28/0.94	36.68/0.98	42.40/0.99	**42.80/0.99**
20	27.65/0.93	31.81/0.95	29.36/0.91	32.68/0.87	33.55/0.94	38.36/0.98	**39.39/0.99**
30	22.71/0.78	28.24/0.91	26.05/0.83	30.21/0.83	31.81/0.95	35.99/0.97	**37.18/0.99**
40	18.23/0.52	25.40/0.85	22.77/0.68	30.01/0.81	30.26/0.93	33.83/0.96	**35.31/0.99**
50	14.61/0.31	22.91/0.77	19.26/0.48	29.32/0.78	28.86/0.91	32.17/0.95	**33.58/0.99**
60	11.74/0.19	20.43/0.69	11.59/0.12	28.75/0.78	27.92/0.89	30.63/0.93	**31.82/0.98**
70	9.43/0.11	18.12/0.59	9.34/0.06	27.04/0.76	26.78/0.86	28.96/0.90	**30.02/0.97**
80	7.56/0.06	15.84/0.51	7.50/0.03	22.25/0.71	25.31/0.81	27.16/0.87	**27.97/0.94**
90	6.07/0.02	13.43/0.43	6.07/0.01	12.88/0.28	22.34/0.70	24.56/0.80	**25.10/0.89**

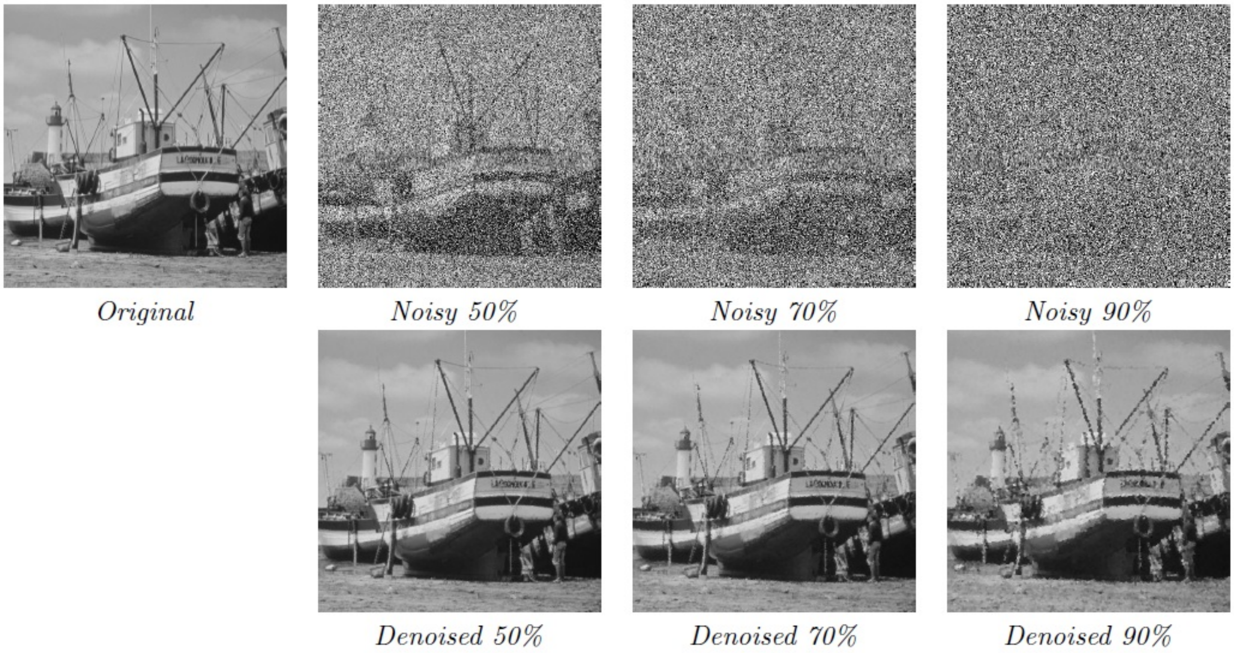
All the results given in the tables were obtained using original Matlab codes of comparison methods on the same noisy images. According to the average PSNR/SSIM results in Tables 4 and 5, we can say that while the proposed method provides better success at different noise levels than MF, DBA, PSMF, ASNLF, MDBUTMF, NAFSMF, DAMF, TVL1, FDS, BPDF, AFMF, AWMF, ACmF, ADKIF (PA) methods, it provides similar results with methods such as ARmF, IAWMF, and DAMRmF. When the average results given for each method are evaluated, we can say that our proposed method is more successful than other methods.



Original *Noisy 50%* *Noisy 70%* *Noisy 90%*

Denoised 50% *Denoised 70%* *Denoised 90%*

Figure 3. Results of the proposed method for Lena image.



Original *Noisy 50%* *Noisy 70%* *Noisy 90%*

Denoised 50% *Denoised 70%* *Denoised 90%*

Figure 4. Results of the proposed method for Boat image.

As another analysis, the performances of the proposed method and other median-based filters for different noise intensities on the Goldhill and Boat image are shown in Figure 6. It shows the visual improvement achieved by our algorithm, moving from low noise density starting from 10% to high noise density ending with 90%. According to the simulation results, the proposed strategy uses a simple concept that yields visually and numerically good results. The average processing times of the proposed and other methods are given in Table

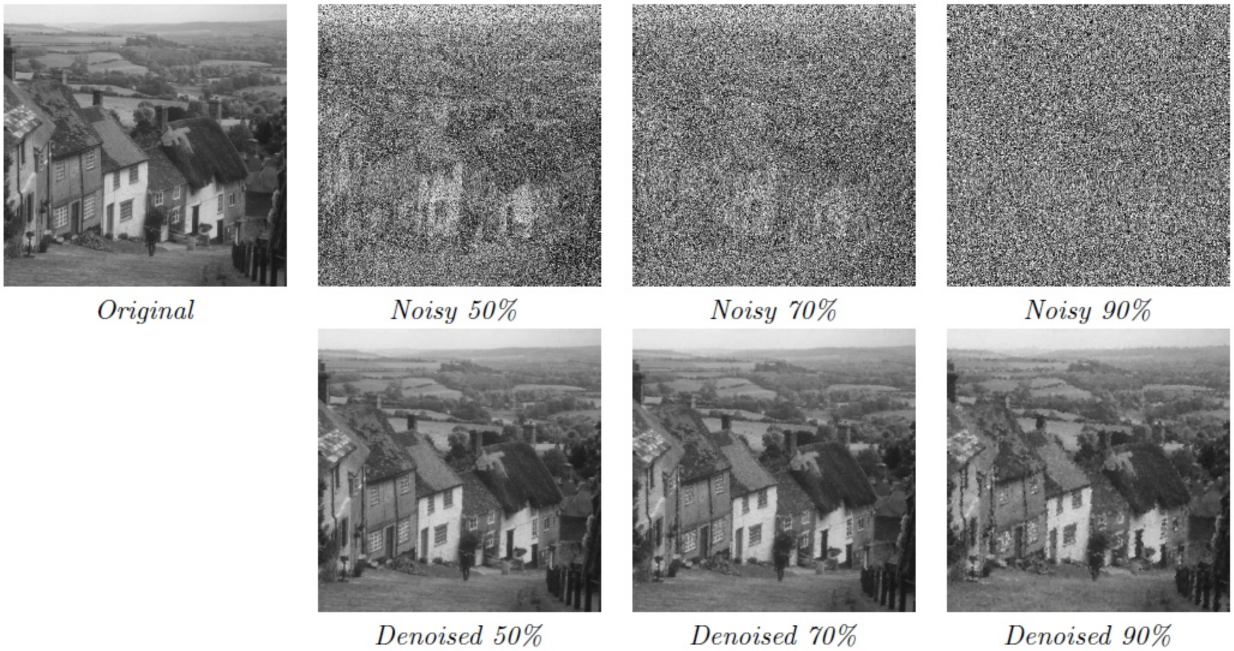


Figure 5. Results of the proposed method for Goldhill image.

Table 4. Average PSNR/SSIM results of the methods for the 20 test images.

Methods	20%	40%	60%	80%	90%	Mean
MF	27.52/0.8405	18.42/0.4592	12.40/0.1300	8.92/0.0301	6.51/0.0131	14.75/0.2945
DBA	34.95/0.9692	29.49/0.8279	25.68/0.6880	23.41/0.5275	19.08/0.5254	26.52/0.7076
PSMF	31.25/0.9348	24.89/0.7621	22.56/0.7124	19.20/0.6520	12.47/0.3136	19.78/0.6750
ASNLF	29.72/0.8602	26.03/0.7592	24.03/0.7504	21.42/0.7312	15.27/0.6529	23.29/0.7509
MDBUTMF	33.22/0.9207	29.15/0.8012	28.40/0.7470	23.56/0.7146	22.40/0.6424	27.35/0.7652
NAFSMF	34.01/0.9510	30.52/0.9017	27.34/0.8297	25.94/0.7410	22.39/0.6229	28.04/0.8092
DAMF	36.45/0.9692	32.79/0.9372	29.70/0.8820	25.92/0.7937	24.43/0.7084	29.86/0.8581
TVL1	29.32/0.8214	27.42/0.7721	24.72/0.6841	21.42/0.5845	15.73/0.3522	23.72/0.6429
FDS	33.42/0.9397	30.50/0.9106	26.28/0.7996	21.90/0.6329	12.48/0.2984	24.91/0.7162
BPDF	33.40/0.9421	31.22/0.8992	28.03/0.8346	25.62/0.7420	10.72/0.2706	25.80/0.7377
AFMF	34.03/0.9445	31.64/0.9172	28.92/0.8612	25.93/0.7610	20.96/0.6263	28.30/0.8220
AWMF	35.19/0.9645	32.78/0.9421	30.05/0.8940	26.72/0.7958	24.71/0.7145	29.89/0.8621
ACmF	36.40/0.9773	33.21/0.9408	30.12/0.8895	26.91/0.8021	24.74/0.7152	30.28/0.8650
ADKIF(PA)	36.78/0.9775	33.24/0.9417	30.08/0.8902	26.95/0.8110	23.96/0.7258	30.20/0.8692
ARmF	37.11/0.9729	33.49/0.9490	30.31/0.8892	27.12/0.8015	24.78/0.7186	30.56/0.8661
IAWMF	36.95/0.9810	33.41/0.9521	30.34/0.8954	27.14/0.8134	25.28/0.7353	30.62/0.8754
DAMRmF	**37.18**/0.9802	33.18/0.9567	30.78/0.8960	**27.51**/0.8145	25.38/0.7384	30.81/0.8772
Proposed	37.13/**0.9917**	**33.53/0.9688**	**30.81/0.9061**	27.41/**0.8427**	**25.41/0.7429**	**30.86/0.8904**

6. Here, we use the computer with Windows 10, 64-bit, Intel Core i5-8250U CPU, 4GB memory and MATLAB R2019a version. According to the results obtained, it is seen that the higher the noise density, the more noisy pixels will be processed, thus increasing the total processing time. It is seen that the proposed method generally has a shorter processing time at different noise intensities.

Table 5. Average PSNR/SSIM results of the methods for the 40 test images.

Methods	20%	40%	60%	80%	90%	Mean
MF	28.32/0.8914	18.49/0.5254	12.30/0.1520	7.63/0.0421	6.04/0.0161	14.55/0.3254
DBA	34.94/0.9690	29.41/0.9021	25.82/0.7967	18.41/0.6597	17.43/0.5181	25.20/0.7691
PSMF	31.24/0.9121	24.46/0.7519	15.40/0.2959	8.05/0.0521	6.03/0.0167	17.04/0.4057
ASNLF	27.03/0.8018	23.51/0.7276	21.92/0.7020	24.90/0.7432	18.27/0.6513	23.13/0.7252
MDBUTMF	34.05/0.9239	28.20/0.7810	26.26/0.8792	23.75/0.7690	21.00/0.6837	26.65/0.8074
NAFSMF	33.10/0.9641	29.49/0.9212	26.43/0.8729	24.35/0.7913	16.03/0.3168	25.88/0.7733
DAMF	37.08/0.9801	32.62/0.9518	29.55/0.9124	26.28/0.8372	23.65/0.7572	29.84/0.8877
TVL1	31.14/0.8824	29.61/0.8392	25.29/0.7580	21.12/0.6245	12.32/0.3107	23.89/0.6830
FDS	34.37/0.9580	30.21/0.9339	26.02/0.8429	21.29/0.6546	18.49/0.5768	26.08/0.7932
BPDF	33.08/0.9602	30.42/0.9125	28.61/0.8722	24.91/0.7904	8.76/0.2041	25.16/0.7479
AFMF	32.82/0.9708	30.19/0.9431	27.48/0.8991	25.20/0.8170	19.85/0.6593	27.11/0.8579
AWMF	36.72/0.9734	34.43/0.9587	31.24/0.9218	27.69/0.8511	24.88/0.7675	30.99/0.8945
ACmF	38.88/0.9829	34.93/0.9614	31.43/0.9256	27.68/0.8517	24.87/0.7678	31.56/0.8979
ADKIF(PA)	38.40/0.9811	34.20/0.9590	30.93/0.9279	27.59/0.8523	25.92/0.7724	31.41/0.8985
ARmF	**39.52**/0.9852	35.40/0.9653	31.77/0.9299	27.79/0.8556	24.89/0.7702	31.87/0.9013
IAWMF	39.28/0.9851	35.16/0.9660	**31.82/0.9319**	28.21/0.8654	25.50/0.7871	31.99/0.9071
DAMRmF	35.41/0.9812	32.87/0.9619	30.94/0.9300	28.12/0.8647	25,46/**0.7894**	30.56/0.9054
Proposed	39.30/**0.9874**	35.21/**0.9669**	31.60/0.9262	**28.86/0.8685**	**26.04**/0.7886	**32.20/0.9075**

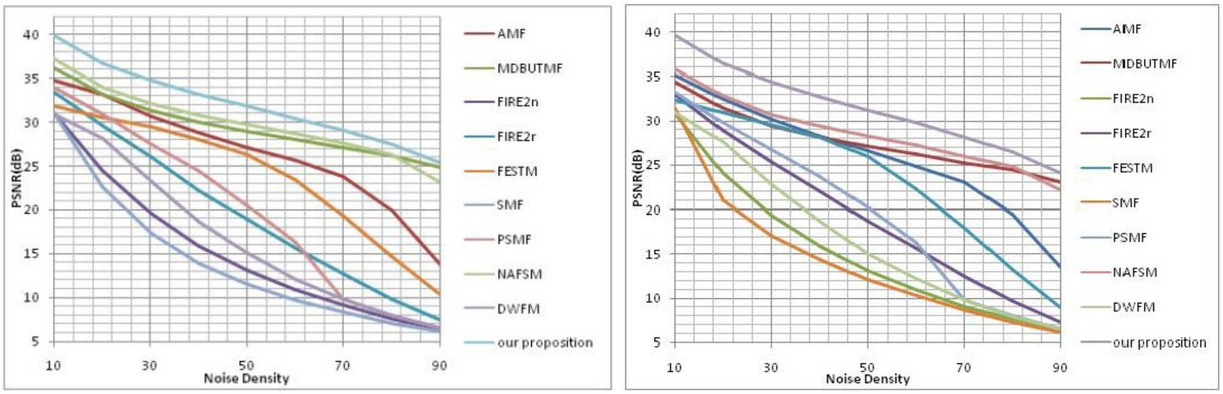


Figure 6. PSNR graphs of Goldhill and Boat images.

4. Conclusions

Salt and pepper noise adversely affects the detailed analysis of the image. For this, it is very important that pixels affected by noise are restored without losing image fine details, especially at high noise density. The proposed algorithm uses a simple concept of detection based on values that can be taken in salt and pepper noise. However, if the original image already has black or white regions, our algorithm solves this problem by selecting the common pixel using the concept of median to restore the corrupted one (in the absence of noiseless pixels). In the second scenario we use a simple averaging filter that produces a pixel very close to the original (in case of noisy pixel detection with no noise pixels). Our concept does not use a variable size window with many parameters to control in the selected window (minimum, maximum and median) used for detection and correction, making the filtration process really simple. Also, the use of larger windows creates a blur effect and

Table 6. Average processing times comparisons for 20 traditional images with other methods (in second).

Noise (%)	TVL1	BPDF	ARmF	ACmF	AWMF	DAMRmF	NAFSM	DAMF	Proposed
10	13.08	8.04	1.27	1.81	16.39	6.06	3.25	1.91	**1.04**
20	12.98	12.93	2.16	3.41	17.09	10.62	5.74	3.39	**2.03**
30	13.01	18.22	**2.85**	4.81	16.27	14.51	8.34	4.72	3.00
40	12.89	23.26	4.19	6.92	16.84	19.02	10.89	6.28	**3.94**
50	12.98	28.53	**4.91**	7.86	17.07	22.66	13.31	7.71	5.08
60	12.91	34.26	6.10	9.80	17.88	27.83	15.84	9.33	**5.89**
70	13.04	39.12	8.39	11.81	18.37	37.84	18.07	11.05	**6.88**
80	12.86	44.16	9.97	14.75	17.77	41.13	20.61	12.31	**8.19**
90	13.02	49.62	12.07	17.57	18.29	52.82	23.49	13.60	**10.12**

produces pixels with a density slightly off the original ones, especially at the edges. In this study, we propose an effective noise removal filter for high density noise level, which can effectively restore the image in terms of quality and speed with less complexity. As a result of experimental studies, it is seen that the proposed algorithm gives successful results without causing any blurring, especially at high noise densities. However, an algorithm can be designed to run in parallel to improve the working time of the proposed method, especially in the preservation of fine details, in images where the noise density is even more intense. Therefore, future studies may focus on parallel design of the algorithm.

Acknowledgment

We would like to express our deep gratitude to Prof. Dr. Ismail Rakip Karas for his support and encouragement.

References

[1] Gonzalez RC, Woods RE. Digital Image Processing, 3rd ed, Prentice Hall: Upper Saddle River, NJ, USA, 2008.

[2] Mafi M, Rajaei H, Cabrerizo M, Adjouadi M. A robust edge detection approach in the presence of high impulse noise intensity through switching adaptive median and fixed weighted mean filtering. IEEE Transactions on Image Processing 2018; 27 (11): 5475-5490 doi: 10.1109/TIP.2018.2857448

[3] Fu B, Zhao X, Song C, Li X, Wang X. A salt and pepper noise image denoising method based on the generative classification. Multimedia Tools and App. 2019; 78 (9): 12043–12053. doi: 10.1007/s11042-018-6732-8

[4] White PR, Baumann O, DeStefano A. On improving the noise reduction algorithms using image segmentation. In: Proceedings of European Signal Processing Conference (EUSIPCO); 2004, pp. 501–504

[5] Ou Q, Zhang H, Li L, Xiong B. Impulse noise removal in two-dimensional electrophoresis images based on dome recognition. In: IEEE 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI); Shenyang; 2015. pp. 440-445

[6] Pratt WK. Median Filtering, Technology Rep, Image Processing Institute University of Southern California: Los Angeles, CA, USA, 1975

[7] Hwang H, Haddad RA. Adaptive median filters: new algorithms and results. IEEE Transactions on Image Processing 1995; 4 (4): 499–502. doi: 10.1109/83.370679

[8] Yin L, Yang R, Gabbouj M, Neuvo Y. Weighted median filters: a tutorial. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 1996; 43 (3): 157–192. doi: 10.1109/82.486465

[9] Ko SJ, Lee YH. Center weighted median filters and their applications to image enhancement. IEEE Transactions on Circuits and Systems 1991; 38 (9): 984–993. doi: 10.1109/31.83870

[10] Pattnaik A, Agarwal S, Chand S. A new and efficient method for removal of high density salt and pepper noise through cascade decision based filtering algorithm. In: 2nd International Conference on Communication, Computing & Security, Procedia Technology, 2012, 6, pp. 108–117

[11] Chan RH, Chung-Wa H, Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. IEEE Transactions on Image Processing 2005; 14 (10): 1479-1485. doi: 10.1109/TIP.2005.852196

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

[13] Wang Z, Zhang D. Progressive switching median filter for the removal of impulse noise from highly corrupted images. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 1999; 46 (1): 78-80. doi: 10.1109/82.749102

[14] Kang CC, Wang WJ. Modified switching median filter with one more noise detector for impulse noise removal. AEU - International Journal of Electronics and Communications 2009; 63 (11): 998-1004. doi: 10.1016/j.aeue.2008.08.009

[15] Akkoul S, Ledee R, Leconge R, Harba R. A new adaptive switching median filter. IEEE Signal Processing Letters 2010; 17 (6): 587-590. doi: 10.1109/LSP.2010.2048646

[16] Faragallah OS, Ibrahem HM. Adaptive switching weighted median filter framework for suppressing salt-and-pepper noise. AEU - International Journal of Electronics and Communications 2016; 70 (8): 1034-1040. doi: 10.1016/j.aeue.2016.04.018

[17] Zhang X, Xiong Y. Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter. IEEE Signal Processing Letters 2009; 16 (4): 295-298. doi: 10.1109/LSP.2009.2014293

[18] Dong Y, Xu S. A new directional weighted median filter for removal of random-valued impulse noise. IEEE Signal Processing Letters 2007; 14 (3): 193–196. doi: 10.1109/LSP.2006.884014

[19] Deivalakshmi S, Sarath S, Palanisamy P. Detection and removal of salt and pepper noise in images by improved median filter. In: IEEE 2011 Recent Advances in Intelligent Computational Systems; 2011, pp. 363-368.

[20] Zhang P, Li F. A new adaptive weighted mean filter for removing salt-and-pepper noise. IEEE Signal Processing Letters 2014; 21 (10): 1280-1283. doi: 10.1109/LSP.2014.2333012

[21] Saleem SA, Razak TA. An effective noise adaptive median filter for removing high density impulse noises in color images. International Journal of Electrical and Computer Engineering (IJECE) 2016; 6 (2): 611-620. doi: 10.11591/ijece.v6i2.pp 611-620

[22] Lu CT, Chou TC. Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter. Pattern Recognition Letters 2012; 33 (10): 1287–1295. doi: 10.1016/j.patrec.2012.03.025

[23] Esakkirajan S, Veerakumar T, Subramanyam AN, PremChand CH. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. IEEE Signal Processing Letters 2011; 18 (5): 287-290. doi: 10.1109/LSP.2011.2122333

[24] Erkan U, Gokrem L. A new method based on pixel density in salt and pepper noise removal. Turkish J. Electr. Eng. Comput. Sci. 2018; 26, 162-171. doi: 10.3906/elk-1705-256

[25] Toh KKV, Isa NAM. Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. IEEE Signal Processing Letters 2010; 17 (3): 281-284. doi: 10.1109/LSP.2009.2038769

[26] Kiani V, Zohrevand A. A fuzzy directional median filter for fixed-value impulse noise removal. In: IEEE 2019 7th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS); Bojnord, Iran, 2019, pp. 1-4.

[27] Karthik B, Kumar TK, Vijayaragavan SP, Sriram M. Removal of high density salt and pepper noise in color image through modified cascaded filter. Journal of Ambient Intelligence and Humanized Computing 2021; 12 (3): 3901-3908. doi: 10.1007/s12652-020-01737-1

[28] Singh V, Dev R, Dhar NK, Agrawal P, Verma NK. Adaptive type-2 fuzzy approach for filtering salt and pepper noise in grayscale images. IEEE Transactions on Fuzzy Systems 2018; 26 (5): 3170-3176. doi: 10.1109/TFUZZ.2018.2805289

[29] Varghese J, Tairan N, Subash S. Adaptive switching non-local filter for the restoration of salt and pepper impulse-corrupted digital images. Arabian Journal for Science and Engineering 2015; 40 (11): 3233-3246. doi: 10.1007/s13369-015-1799-2

[30] Zhang C, Wang K. A switching median–mean filter for removal of high-density impulse noise from digital images. Optik 2015; 126 (9-10): 956-961. doi: 10.1016/j.ijleo.2015.02.085

[31] Balasubramanian G, Chilambuchelvan A, Vijayan S, Gowrison G. Probabilistic decision based filter to remove impulse noise using patch else trimmed median. AEU - International Journal of Electronics and Communications 2016; 70 (4): 471-481. doi: 10.1016/j.aeue.2016.01.013

[32] Erkan U, Gökrem L, Enginoğlu S. Different applied median filter in salt and pepper noise. Computers & Electrical Engineering 2018; 70, 789-798. doi: 10.1016/j.compeleceng.2018.01.019

[33] Varatharajan R, Vasanth K, Gunasekaran M, Priyan M, Gao XZ. An adaptive decision based kriging interpolation algorithm for the removal of high density salt and pepper noise in images. Computers & Electrical Engineering 2018; 70, 447-461. doi: 10.1016/j.compeleceng.2017.05.035

[34] Enginoglu S, Erkan U, Memis S. Pixel similarity-based adaptive riesz mean filter for salt-and-pepper noise removal. Multimedia Tools and Applications 2019; 78 (24): 35401-35418. doi: 10.1007/s11042-019-08110-1

[35] Enginoğlu S, Erkan U, Memiş S. Adaptive cesáro mean filter for salt-and-pepper noise removal. El-Cezeri 2020; 7, 304-314. doi: 10.31202/ecjse.646359.

[36] Erkan U, Thanh DNH, Enginoğlu S, Memiş S. Improved adaptive weighted mean filter for salt-and-pepper noise removal. In: ICECCE 2020 International Conference on Electrical, Communication, and Computer Engineering, 2020, pp. 1-5.

[37] Memiş S, Erkan U. Different adaptive modified riesz mean filter for high-density salt-and-pepper noise removal in grayscale images. Avrupa Bilim ve Teknoloji Dergisi 2021; 359-367. doi: 10.31590/ejosat.873312

[38] Erkan U, Enginoğlu S, Thanh DNH, Hieu LM. Adaptive frequency median filter for the salt and pepper denoising problem. IET Image Processing 2020; 14 (7): 1291-1302. doi: 10.1049/iet-ipr.2019.0398

[39] Aghajarian M, McInroy JE, Wright CHG. Salt-and-pepper noise removal using modified mean filter and total variation minimization. Journal of Electronic Imaging 2018; 27 (1), doi: 10.1117/1.JEI.27.1.013002

[40] Thanh DNH, Surya PVB, Thanh LT. Total variation l1 fidelity salt-and-pepper denoising with adaptive regularization parameter. In: IEEE 2018 5th NAFOSTED Conference on Information and Computer Science (NICS); Ho Chi Minh City, 2018, pp. 400–405.

[41] Xing Y, Xu J, Tan J, Li D, Zha W. Deep CNN for removal of salt and pepper noise. IET Image Processing 2019; 13 (9): 1550-1560. doi: 10.1049/iet-ipr.2018.6004

[42] Jin L, Zhang W, Ma G, Song E. Learning deep CNNs for impulse noise removal in images. Journal of Visual Communication and Image Representation 2019; 62, 193-205. doi: 10.1016/j.jvcir.2019.05.005

[43] Sadrizadeh S, Zarmehi N, Kangarshahi EA, Abin H, Marvasti, F. A fast iterative method for removing impulsive noise from sparse signals. IEEE Transactions on Circuits and Systems for Video Technology 2021; 31 (1): 38-48. doi: 10.1109/TCSVT.2020.2969563

[44] Sen AP, Rout, NK. Improved probabilistic decision-based trimmed median filter for detection and removal of high-density impulsive noise. IET Image Processing 2020; 14 (17): 4486-4498. doi: 10.1049/iet-ipr.2019.1240

[45] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 2004; 13 (4): 600-612. doi: 10.1109/TIP.2003.819861

[46] Asuni N, Giachetti A. TESTIMAGES: a large-scale archive for testing visual devices and basic image processing algorithms. In: Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference, Giachetti, A, Ed, The Eurographics Association, 2014.