

Article

Salt and Pepper Noise Removal Method Based on a Detail-Aware Filter

Hu Liang ¹, Na Li ² and Shengrong Zhao ^{1,*}

¹ College of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250023, China; lianghu@qlu.edu.cn

² Shandong Computer Science Center (National Supercomputing Center in Jinan), Jinan 250101, China; lina@sdas.org

* Correspondence: zhaoshengrong@qlu.edu.cn

Abstract: The median-type filter is an effective technique to remove salt and pepper (SAP) noise; however, such a mechanism cannot always effectively remove noise and preserve details due to the local diversity singularity and local non-stationarity. In this paper, a two-step SAP removal method was proposed based on the analysis of the median-type filter errors. In the first step, a median-type filter was used to process the image corrupted by SAP noise. Then, in the second step, a novel-designed adaptive nonlocal bilateral filter is used to weaken the error of the median-type filter. By building histograms of median-type filter errors, we found that the error almost obeys Gaussian–Laplacian mixture distribution statistically. Following this, an improved bilateral filter was proposed to utilize the nonlocal feature and bilateral filter to weaken the median-type filter errors. In the proposed filter, (1) the nonlocal strategy is introduced to improve the bilateral filter, and the intensity similarity is measured between image patches instead pixels; (2) a novel norm based on half-quadratic estimation is used to measure the image patch- spatial proximity and intensity similarity, instead of fixed L_1 and L_2 norms; (3) besides, the scale parameters, which were used to control the behavior of the half-quadratic norm, were updated based on the local image feature. Experimental results showed that the proposed method performed better compared with the state-of-the-art methods.

Keywords: salt and pepper; median-type filter; bilateral filter; nonlocal



Citation: Liang, H.; Li, N.; Zhao, S. Salt and Pepper Noise Removal Method Based on a Detail-Aware Filter. *Symmetry* **2021**, *13*, 515. <https://doi.org/10.3390/sym13030515>

Academic Editor: Tomohiro Inagaki

Received: 27 February 2021

Accepted: 18 March 2021

Published: 21 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Detail restoration is a challenging problem that is necessary for some image processing tasks, such as feature extraction, object identification, and pattern recognition. The impulse noise remains in the process of image acquisition and transmission [1,2], which is an inevitable and unwanted phenomenon. The salt and pepper (SAP) noise is a type of impulse noise where the corrupted pixel takes either the maximum or minimum gray value. This noise appears as white and black pixels in the corrupted image [3–5]. Moreover, the challenge in detail restoration is further amplified when the images are corrupted by heavy SAP noise owing to the significant destruction of information detail.

In this work, a two-step SAP noise removal method is proposed. The aim of this method is to study the denoising result of the median-type filter further, and propose a novel method to improve the visual quality. First, by analyzing the error of the median-type filter, it was observed to adhere to a Gaussian–Laplacian mixture distribution statistically. Following this observation, in the second step, an adaptive nonlocal bilateral filter has been proposed to utilize the nonlocal feature and bilateral filter to recover the result of median-type filters. In the proposed filter, the difference between the image patch is measured by a modified version of the adaptive norm proposed in [6], which is different from the traditional methods. Moreover, the scale parameters, which were used to control

the behavior of the adaptive norm, were updated based on the local feature for higher estimation accuracy.

The contributions of this paper are summarized as follows:

- (1) By analyzing the error of the median-type filter statistically, it is found that the error is almost a Gaussian–Laplacian mixture distribution. Therefore, a two-step noise removal method is designed to remove the SAP noise.
- (2) A novel adaptive non-local bilateral filter is proposed to recover the median-type filtered result. Owing to the drawbacks of traditional bilateral filter, a nonlocal operator is used to extract image patches, and the adaptive norm is used to measure the spatial proximity and intensity similarity between the patches.
- (3) We propose a method to calculate the scale parameters in the adaptive norm. Using this strategy, the context information can be utilized to make the norm adapt to the patch feature.

2. Related Work

There are several methods proposed to reconstruct images corrupted by SAP noise in the past. The research for nonlinear filter is actively pursued in the field of SAP noise removal. Median-type filters [7–10] are the most popular nonlinear filters. Such as the standard median filter (MF), which is one of the most popular nonlinear filters. MF convolves a moving window with determined size over the image. If the pixel at the center of the window is noisy, its value is replaced with the current window's median value. However, the noise-free pixels are treated similar to the noisy pixels when these filters are used to restore the corrupted image. These pixels are also replaced with estimated values, which lead to artifacts and blurring. This results in (1) the distortion of original image details and (2) the increase in the computation, especially in the case of low signal–noise ratio.

2.1. Switching Filters

To solve this problem, the switching median-type filters are proposed. The main idea of switching filters is to use a switching process to select the optimal output in the noise detection step or correction step. In [11], first, the authors propose impulse noise detection methods based on evidential reasoning. Then, they design an adaptive switching median filtering which could adaptively determine the size of filtering window according to detection results. The quaternion switching vector median filter, introduced in [12], detects impulse noise based on a quaternion-based color distance and local reachability density. The quaternion-based color distance is used to calculate the local density of a color pixel. Chanu and Singh [13] also propose a switching vector median filter based on quaternion to remove impulse noise, which contains two stages. In the first stage, they use a rank strategy to determine whether the central pixel of the filtering window is noisy pixel or not. Then, in the second stage, the probably corrupted candidate is re-confirmed by using four Laplacian convolution kernels. The noisy pixel is processed by a switching vector median filter based on the quaternion distance.

2.2. Decision Filters

The decision-based method has attracted significant attention for removing SAP noise. Decision filters assume that noisy pixels have a value of 0 or 255 while noise-free pixels have a value between them. As presented in [14], a new filter is established on decision-based filters. The method contains noise detection and restoration. In restoration phase, the uncorrupted pixels keep unchanged and corrupted pixels are interpolated from surrounding uncorrupted pixels using a two-dimensional scattered data interpolation, named natural neighbor Galerkin method. In [15], the noise-free pixels are also left unprocessed and the noisy pixels are replaced with the Kriging interpolation value. The noisy pixels are interpolated from the noise-free pixels in a confined neighborhood. The weights of contrib-

tors are calculated using semi-variance between the corrupted pixels and uncorrupted pixels. Besides, an adaptive window size is used for the increasing noise densities.

2.3. Fuzzy Filters

Toh and Isa [16] propose the noise adaptive fuzzy switching median filter (NAFSMF), which is a hybrid between the simple adaptive median filter [17] and the fuzzy switching median filter [18]. NAFSMF contains two stages: noise detection and removal. In the detection stage, the histogram of the corrupted image is utilized to identify noisy pixels. In the removal stage, fuzzy reasoning is employed to deal with uncertainty and design a correction term to estimate noisy pixels. In [19], the authors first improve the maximum absolute luminance difference (ALD) method for detecting noisy pixels more accurately, which helps to classify the pixels into three categories: uncorrupted ones, lightly corrupted ones, and heavily corrupted ones. To restore corrupted pixels, a distance relevant adaptive fuzzy switching weighted mean filter is implemented to remove noise. Afterward, the uncertainty present in the extracted local information as introduced by noise is handled via fuzzy reasoning [20,21].

2.4. Morphological Filters

Morphological filters are non-linear and it can modify the geometrical features locally [22,23]. Mediated morphological filters, introduced in [23], are a combination of median filtering and classical gray-scale morphological operators. After, the mediated morphological filters are applied to remove noise from adult and fetal electrocardiogram (ECG) signal [24,25] and medical images [26]. In [24], mediated morphological filters are used as an efficient preprocessing step to deal with adult and fetal ECG signals. Such a strategy could suppress noise effectively and show low sensitivity to the changes of the structuring element's length. Compared with [24], the method presented in [25] further concludes the preprocessing result by employing a morphological background normalization. In [26], the mediated morphological filter ability is verified in removing SAP noise, Gaussian noise and Speckle noise from medical images. Compared with the weighted MF, classical MF, and linear filter, the mediated morphological filter shows better performance, even in high abnormal condition of each kind of noise.

2.5. Cascade Filters

The cascade methods use the idea of combining different filters to improve the restoration quality. Such as, the authors of [27] combine the switching adaptive median with fixed weighted mean filter (SAMFWMF). This filter contains switching adaptive median filtering and fixed weighted mean filtering with additional shrinkage window. This filter can achieve optimal edge detection and preservation. In [28], a decision-based median filter is combined with asymmetric trimmed mean filter. In this method, the pixel whose value is equal to 0 or 255 is replaced with the median of the moving window; otherwise, it is replaced with the mean of the window. Raza and Sawant [29] combine a decision-based median filter with a modified decision-based partially trimmed global mean filter (DBPTGMF). Esakkirajan et al. [30] combine a decision-based median filter with a modified decision-based unsymmetrical trimmed median filter (MDBUTMF). Both DBPTGMF and MDBUTMF contain two stages. The first stage is common for both approaches: detect the noisy pixels and replace them with the window's median. Two main differences exist in the second stage: (I) when detecting the corrupted pixels in the window, if all the pixels in the window (except for the central corrupted pixel) are 0 s (or 1 s), DBPTGMF replaces the corrupted pixel with 0 (or 1). On the other hand, MDBUTMF replaces the corrupted pixel with the window's mean. (II) If the pixels in the window are a combination of 0 s and 1 s, DBPTGMF replaces the corrupted pixel with the mean of the window. However, the corrupted pixel will be replaced with the median of the window if at least one pixel in the window is different from 0 or 1. For both cases, MDBUTMF replaces the corrupted pixel with the median of the window.

2.6. Nonlocal Means Filter and Bilateral Filter for SAP Noise Removal

Most of the existing SAP noise removal methods are pixel-based methods, which consider the image pixels in a fixed local region only and ignore the image self-similar information. Thus, the image nonlocal structure or the texture structure cannot be preserved properly. To solve this problem, Wang et al. [31] propose the iterative nonlocal means (INLM) filter to remove SAP noise. In this method, switching median filter is first used to mark the pixels as noisy or noise-free pixels and performs filtering on the noisy pixels only. Then, an iterative nonlocal means framework is used to estimate the noisy pixels. In [31], they iteratively exploited the nonlocal similarity feature of the image. They also obtain higher accuracy by updating the similarity weights and the estimated values simultaneously. The bilateral filter is a widely used method for Gaussian noise suppression, and it is barely discussed for SAP noise removal. Veerakumar et al. [32] propose an adaptive bilateral filter by modifying the spatial proximity-based and intensity similarity-based Gaussian function. In the intensity similarity-based Gaussian function, the difference between different pixels is measured by the L_2 norm, which penalizes the high-frequency component, and this may blur the edge and texture.

3. Error Analysis and Bilateral Filter

3.1. Error Analysis

In this work, we focused on the statistical characteristics of the results other than the method itself. Based on this idea, we first add salt and pepper noise with various intensities to the image and then remove the noise using the median type filter (i.e., MF, NAFSM)

$$\hat{I}_{mf} = \text{MedianFilter}(I) \quad (1)$$

where \hat{I}_{mf} denotes the filtered image and I is the image corrupted by the SAP noise. Then, we calculated the error image

$$I_{err} = \hat{I}_{mf} - I_{ori} \quad (2)$$

where I_{err} is the error image between the filtered image \hat{I}_{mf} and the original image I_{ori} .

Then, we compute the normalized histogram for the error image. Take the image ‘‘Lena’’ as an example, Figure 1a is the error image between I and the original image I_{ori} , Figure 1b is the histogram for Figure 1a, Figure 1c is the error image between the filtered image \hat{I}_{mf} and the original image I_{ori} . To verify our hypothesis, we repeated these experiments in different images with different sizes and noise intensities. Furthermore, the histogram of the error of these images is shown in Figure 1d. It can be seen that the curves corresponding to different images are all very close to the red curve (i.e., Gaussian distribution defined in Equation (3)) at both ends, while close to the black curve (i.e., Laplacian distribution defined in Equation (4)) near the peak area. It means that after prefiltering, the noise remained in the image obeys a Gaussian–Laplacian mixture distribution.

If r obeys Gaussian distribution, i.e., $r \sim N(\mu_G, \sigma_G)$, the distribution function is defined as

$$\rho_G(r) = \frac{1}{\sqrt{2\pi}\sigma_G} \exp\left(-\frac{(r - \mu_G)^2}{2\sigma_G^2}\right) \quad (3)$$

If t obeys Laplacian distribution, i.e., $t \sim La(\mu_L, \sigma_L)$, the distribution function is defined as

$$\rho_L(t) = \frac{1}{2\sigma_L} \exp\left(-\frac{|t - \mu_L|}{\sigma_L}\right) \quad (4)$$

where, μ_G and μ_L are position parameter, σ_G and σ_L are scale parameters.

Intuitively, we deduce that the image obtained by the median-type filter is similar to an image with mixture noise, which can be written as

$$\hat{I}_{mf} = I_{ori} + n_{G\&L} \quad (5)$$

This phenomenon gives us an inspiration that a better denoising performance would be achieved by successfully designing a Gaussian–Laplacian mixed noise filter and introducing it into a SAP noise removal problem. In this paper, this question will be discussed and satisfactory results are obtained by applying the modified bilateral filter into salt-and-pepper noise removal.

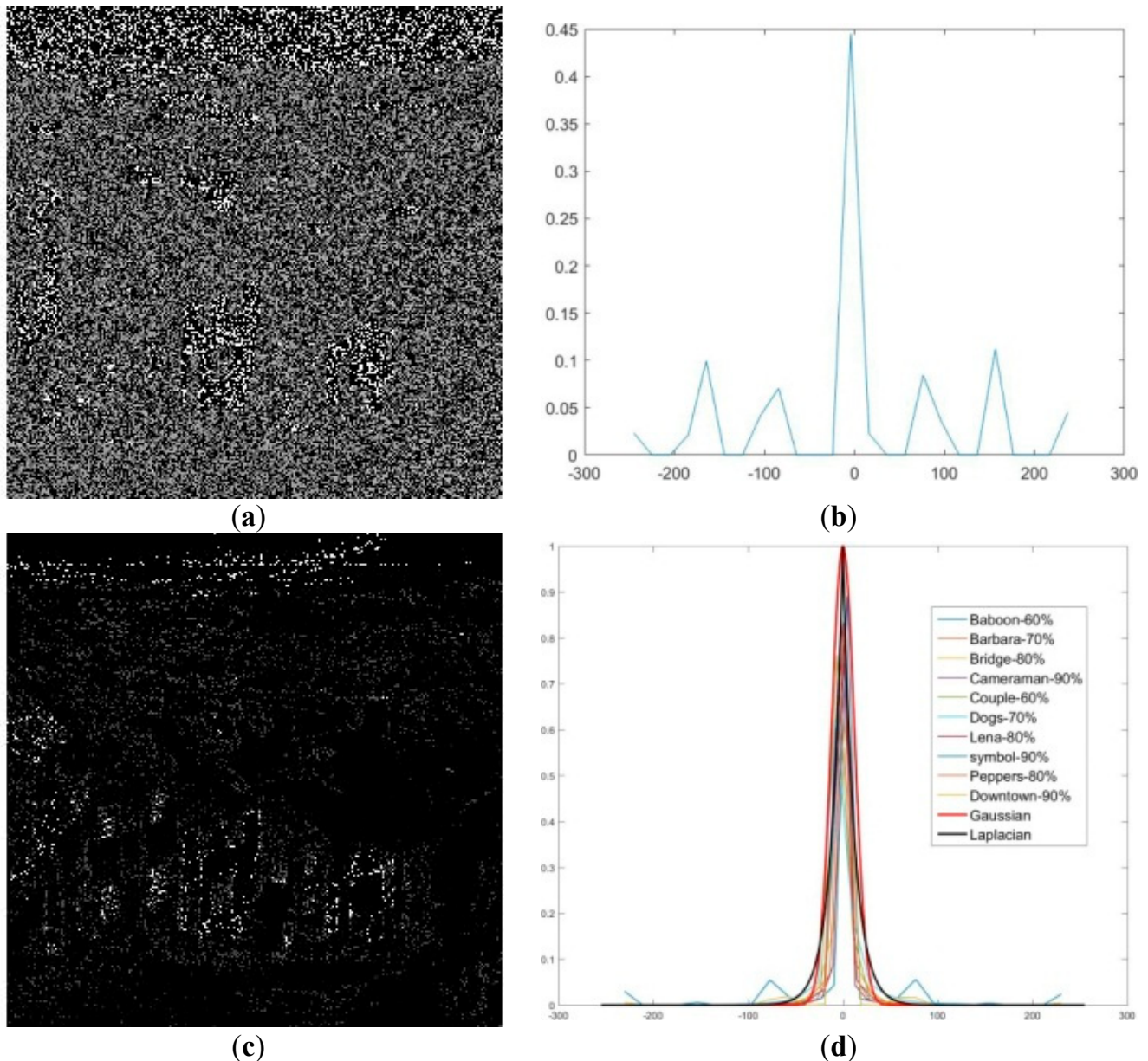


Figure 1. Error image and the corresponding histogram. (a) Error image between I and I_{ori} for image “Lena”; (b) histogram of (a); (c) error image between I_{ori} and \hat{I}_{mf} for the image “Lena”; (d) histograms of the error image for 10 images.

3.2. Bilateral Filter

Bilateral filter [33,34] is a kind of nonlinear filter aimed at edge preservation, since it simultaneously considers spatial proximity and the intensity similarity between image

pixels. Mathematically, the corrupted image pixel can be estimated by the weighted average of all the neighborhood pixels as

$$x(i, j) = \frac{\sum_{(k,l) \in B(i,j)} \omega_d(i, j, k, l) \omega_r(i, j, k, l) I(k, l)}{\sum_{(k,l) \in B(i,j)} \omega_d(i, j, k, l) \omega_r(i, j, k, l)} \quad (6)$$

where $x(i, j)$ denotes the estimated image pixel located at (i, j) , $B(i, j)$ denotes the patch whose center pixel is located at (i, j) , $I(k, l) \in B(i, j)$, ω_d and ω_r are both Gaussian functions to measure the spatial proximity and the intensity similarity, respectively. Formally, these functions can be written as

$$\omega_d(i, j, k, l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2}\right) \quad (7)$$

$$\omega_r(i, j, k, l) = \exp\left(-\frac{\|I(i, j) - I(k, l)\|_2^2}{2\sigma_r^2}\right) \quad (8)$$

where, σ_d and σ_r are smoothing parameters.

4. Proposed Two-Step Algorithm

This section presents a detailed explanation of the proposed two-step SAP noise removal algorithm. In the first step, a median-type filter (i.e., NAFSMF) was used to process the corrupted image. Then, in the second step, a novel-designed adaptive nonlocal bilateral (ANB) filter is used to weaken the error of the median-type filter given that it is found to have a Gaussian-like distribution statistically.

4.1. NAFSMF for Preprocessing

In this stage, the noisy image histogram is utilized so as to estimate the noisy pixels in the image corrupted by SAP noise. The local maximum method [18] is first used to detect the noisy pixels. By using this method, it can avoid mistaking noise free intensities in the noisy image histogram for the noisy intensities in the case of low noise intensity. The local maximum is the first peak encountered when traversing the noisy image histogram in a particular direction. The search is started from both ends of the histogram and directed towards the center of the histogram. Two noise intensities are found and used to identify possible noisy pixels in the image. The two local maximums are denoted as N_{Salt} and N_{Pepper} , respectively. A noise mask M is designed as follows to mark the location of noisy pixels

$$M(i, j) = \begin{cases} 0, & I(i, j) = N_{Salt} \text{ or } I(i, j) = N_{Pepper} \\ 1, & \text{otherwise} \end{cases} \quad (9)$$

where $I(i, j)$ records the gray value at point (i, j) . $M(i, j) = 1$ denotes that the point (i, j) in I is noise-free; otherwise, $M(i, j) = 0$ denotes that the point (i, j) in I is a noisy point.

To calculate the noisy point $I(i, j)$, first, define a search window $W_{2s+1}(i, j)$ with size $(2s + 1) \times (2s + 1)$

$$W_{2s+1}(i, j) = \{I(i + m, j + n) | m, n \in (-s, s)\} \quad (10)$$

Then, $T_{2s+1}(i, j)$ is used to count the number of noise-free pixels in $W_{2s+1}(i, j)$

$$T_{2s+1}(i, j) = \sum_{m, n \in (-s, s)} N(i + m, j + n) \quad (11)$$

If $T_{2s+1}(i, j) > 1$, the value of the noisy pixel is calculated using

$$I_{Med}(i, j) = \text{median}\{I(i + m, j + n) | M(i + m, j + n) = 1; m, n \in (-s, s)\} \quad (12)$$

If $T_{2s+1}(i, j) < t$, there is not enough noise-free pixels in the current window. In this case, $s + 1 \rightarrow s$ until $T_{2s+1}(i, j) > 1$. Finally, y is used to denote the filtering result and the central pixel $y(i, j)$ is calculated using the equation

$$y(i, j) = (1 - F(i, j))I(i, j) + F(i, j)I_{Med}(i, j) \tag{13}$$

where $F(i, j)$ is a fuzzy membership function that is defined as

$$F(i, j) = \begin{cases} 0, & D(i, j) < T_1 \\ \frac{D(i, j) - T_1}{T_2 - T_1} & T_1 < D(i, j) < T_2 \\ 1 & D(i, j) > T_2 \end{cases} \tag{14}$$

In Equation (14), $D(i, j)$ represents the local information defined as the maximum gray value difference in a 3×3 window, and it is defined as

$$D(i, j) = \max\{|I(i + k, j + l) - I(i, j)|, k, l \in (-2, 2), (i + k, j + l) \neq (i, j)\} \tag{15}$$

4.2. ANB Filter to Improve Result

The bilateral filter has local characteristics due to considering the pertinence of pixels to estimate noisy pixels. The pertinence will be destroyed when encountering high noise intensity, thus the denoising effect will be greatly reduced. To address this problem, we first introduce the nonlocal strategy to the bilateral filter, and a new function to measure the patch intensity similarity is obtained,

$$\bar{\omega}_r(i, j, k, l) = \exp\left(-\frac{\sum_{\substack{y(p, q) \in \Theta(i, j) \\ y(r, s) \in \Theta(k, l) \\ \Theta(i, j), \Theta(k, l) \subset \Omega}} (y(p, q) - y(r, s))^2}{\beta}\right) \tag{16}$$

where, Ω represents the search window, $\Theta(i, j)$ represents the patch centered by $y(i, j)$, $\Theta(k, l)$ represents the patch centered by $y(k, l)$, $\beta > 0$ is a model parameter.

Suppose the vector $v \in R^n$, the Lp norm of v is defined as $\|v\|_p = \left(\sum_{i=1}^n |v_i|^p\right)^{\frac{1}{p}}$. For the Gaussian noise and Laplacian noise, L₂ norm is an excellent choice. However, the L₂ norm is outliers, thus, the L₂ norm does not perform as well as the L₁ norm in preserving sharp edges. Moreover, when using Equations (7) and (16) based method to deal with the preprocessing result, there still exists some problems, such as, (1) all pixels need to be estimated; (2) all pixels in the search window contribute to the estimation; and (3) fixed L₂ norm used in Equation (16) is sensitive to high frequency information, which may blur the image edge. Based on the above analysis, we further modified the spatial proximity-based function and the intensity similarity-based function as

$$\hat{\omega}_d(i, j, k, l) = \begin{cases} \exp\left(-\frac{(i-k)^2 + (j-l)^2}{\alpha}\right) \\ 0 \end{cases} \tag{17}$$

$$\hat{\omega}_r(i, j, k, l) = \begin{cases} \sum_{\substack{y(p, q) \in \Theta(i, j) \\ y(r, s) \in \Theta(k, l) \\ \Theta(i, j), \Theta(k, l) \subset \Omega}} \varphi(y(p, q) - y(r, s), a) \\ \exp\left(-\frac{\quad}{\beta}\right) \\ 0 \end{cases} \tag{18}$$

where, Ω represents the search window, $\Theta(i, j)$ represents the patch centered by $y(i, j)$, $\Theta(k, l)$ represents the patch centered by $y(k, l)$, $\alpha > 0$ and $\beta > 0$ are model parameters, $\varphi(t, a)$ is a measure function defined by $\varphi(t, a) = a\sqrt{a^2 + t^2} - a^2$ with scale parameter $a > 0$. $\varphi(t, a)$ has

self-adaptability of mimicking L_1 and L_2 norms, and the parameter a controls the transition from L_1 norm to L_2 norm. Larger a denotes a larger range of difference values that can be discriminated by the L_2 norm. In the two patches, $\Theta(i, j)$ and $\Theta(k, l)$, the difference $y(p, q) - y(r, s)$ may be variational owing to the location. Thus, using the same a to measure the difference is not an excellent choice. In this work, the parameter a is computed based on the local image feature

$$a_{p,q,r,s} = \frac{1}{(y(p, q) - y(r, s))^2 + \varepsilon} \tag{19}$$

where $\varepsilon > 0$, which is used to avoid making the dividend zero. Detail analysis about $a_{p,q,r,s}$ is presented in Section 5.

Thus, by substituting a with Equation (19) in Equation (18), we can obtain new functions for the distance in intensity space

$$\hat{\omega}_r(i, j, k, l) = \begin{cases} \frac{\sum_{\substack{y(p,q) \in \Theta(i,j) \\ y(r,s) \in \Theta(k,l)}} \psi(y(p,q)-y(r,s), a_{p,q,r,s})}{\beta} \\ 0 \end{cases} \tag{20}$$

where $\psi(t, a_{p,q,r,s}) = a_{p,q,r,s} \sqrt{a_{p,q,r,s}^2 + t^2} - a_{p,q,r,s}^2$.

Based on Equations (17) and (20), the ANB filter is proposed and defined as

$$\hat{x}(i, j) = \frac{\sum_{(k,l) \in \Theta(i,j)} \hat{\omega}_d(i, j, k, l) \hat{\omega}_r(i, j, k, l) y(k, l)}{\sum_{(k,l) \in \Theta(i,j)} \hat{\omega}_d(i, j, k, l) \hat{\omega}_r(i, j, k, l)} \tag{21}$$

In the following section, we will discuss the advantages of the adaptive norm compared with L_1 and L_2 norms in theory. The experimental comparisons are also shown in Section 6.

4.3. Proposed Two-Stage Noise Removal Algorithm

In this subsection, we summarized the whole salt and pepper noise removal algorithm. The whole algorithm contains two stages: (1) median-type filter and (2) the proposed ANB filter. The proposed algorithm is described as Algorithm 1:

Algorithm 1: Proposed two-stage noise removal algorithm

Input: noisy image I , α , β ;

First stage

1. Obtain M using Equation (9);
2. $1 \rightarrow s$

Obtain $W_{2s+1}(i, j)$ and $T_{2s+1}(i, j)$;

Do $s \rightarrow s + 1$ until $T_{2s+1}(i, j) > 1$;

3. Calculate $I_{Med}(i, j) = \text{median}\{I(i + m, j + n) | m, n \in (-s, \dots, 0, \dots, s)\}$;
4. Calculate $y = (1 - F(i, j))I(i, j) + F(i, j)I_{Med}(i, j)$ using Equations (14) and (15);

Second stage

5. Calculate the spatial proximity using

$$\hat{\omega}_d(i, j, k, l) = \begin{cases} \exp\left(-\frac{(i-k)^2 + (j-l)^2}{\alpha}\right) \\ 0 \end{cases}$$

6. Calculate the scale parameters using

$$a_{p,q,r,s} = \frac{1}{(y(p, q) - y(r, s))^2 + \epsilon}$$

7. Calculate the intensity similarity using

$$\hat{\omega}_r(i, j, k, l) = \begin{cases} \frac{\sum_{\substack{y(p,q) \in \Theta(i,j) \\ y(r,s) \in \Theta(k,l)}} \psi(y(p,q) - y(r,s), a_{p,q,r,s})}{\beta} \\ 0 \end{cases}$$

8. Obtain the de-noised result

$$x(i, j) = \frac{\sum_{(k,l) \in \Theta(i,j)} \hat{\omega}_d(i, j, k, l) \hat{\omega}_r(i, j, k, l) y(k, l)}{\sum_{(k,l) \in \Theta(i,j)} \hat{\omega}_d(i, j, k, l) \hat{\omega}_r(i, j, k, l)}$$

Output: de-noised image x

5. Efficiency Analysis

One of the main contributions of this manuscript is the establishment of the ANB filter, which is based on the improved functions ($\hat{\omega}_d$ and $\hat{\omega}_r$) with regard to the spatial proximity and the intensity similarity, respectively. The efficiency analysis mainly includes three aspects: (1) the error analysis, which provides an explanation to add the following ANB step; (2) the weight analysis, which shows the efficiency of the introduction of $\hat{\omega}_d$ and $\hat{\omega}_r$ into the ANB filter; and (3) the norm choice analysis, which presents the efficiency of using the adaptive norm to improve $\hat{\omega}_r$. The error analysis has been presented detailed in Section 2, thus, in this section, the efficiency analysis mainly includes two parts: weight analysis and norm choice.

5.1. Weight Analysis

Two improved weight functions were designed in the proposed filter. The two functions were based on spatial proximity and intensity similarity. The efficiency of using the two weights $\hat{\omega}_d$ and $\hat{\omega}_r$ was analyzed in three special situations as follows:

Case 1: Considering Equations (17) and (20), if the contributed pixel $y(k, l)$ is corrupted, $\hat{\omega}_d = (i, j, k, l) = 0$ and $\hat{\omega}_r = (i, j, k, l) = 0$. To this end, the pixel $y(k, l)$ has no effect on estimating $y(i, j)$. This design can avoid misestimation owing to the corrupted points.

Case 2: If the contributed pixel $y(k, l)$ is noise-free in the smooth region, but $y(i, j)$ in an area where the gray value changes dramatically (i.e., edge region), the difference between the two pixels is large. Correspondingly, the difference between the patches centered by

$y(k, l)$ and $y(i, j)$ may be large. In this case, the variable $\hat{\omega}_r$ is close to 0, and the value of $\hat{\omega}_d$ is small. Thus, it is a double insurance to keep the weight of $y(k, l)$ is small enough when computing for $y(i, j)$.

Case 3: If both $y(k, l)$ and $y(i, j)$ are in the smooth region, a small difference is shown among the image pixels. In this situation, the difference between the patches tends to be 0 infinitely. Thus, the weight $\hat{\omega}_r$ approaches 1, and only the weight $\hat{\omega}_d$ works. Here, the proposed filter works as a Gaussian filter.

5.2. Norm Choice

The proposed norm $\psi(t, a_{p,q,r,s}) = a_{p,q,r,s} \sqrt{a_{p,q,r,s}^2 + t^2} - a_{p,q,r,s}^2$ tends to utilize the advantages of L₁ norm and L₂ norm, and it has a good adaptivity to the local information by avoiding to set the threshold value to control the selection of the norm compared with other estimators (i.e., the Huber norm, the Leclerc norm, and the Lorentzian norm).

From Equation (19), it is noted that when the difference between $y(p, q)$ and $y(r, s)$ is small, the value of $a_{p,q,r,s}$ is large. Here, the modified adaptive norm contains a range where it performs similar to the L₂ norm as large as possible. Conversely, $a_{p,q,r,s}$ is small, and the modified adaptive norm performs similar to the L₁ norm, which can remove noise and preserve edges. Thus, the proposed adaptive norm can work as L₁ norm or L₂ norm automatically according to the local features.

In the following equation,

$$\psi(t) = a_{p,q,r,s} \sqrt{a_{p,q,r,s}^2 + t^2} - a_{p,q,r,s}^2. \quad (22)$$

where $a_{p,q,r,s}$ is used to control the scope of the linear behavior as shown in Figure 2. This norm is shown in Figure 2 with different $a_{p,q,r,s}$ -values where it can be seen that the adaptive norm gets closer to the L₁ norm when the sale parameter tends to zero. Specifically, larger sale parameter results in a larger range of error values that can be discriminated by the linear influence function. Moreover, in $\psi(t)$, the sale parameter controls the transition from L₁ norm to L₂ norm.

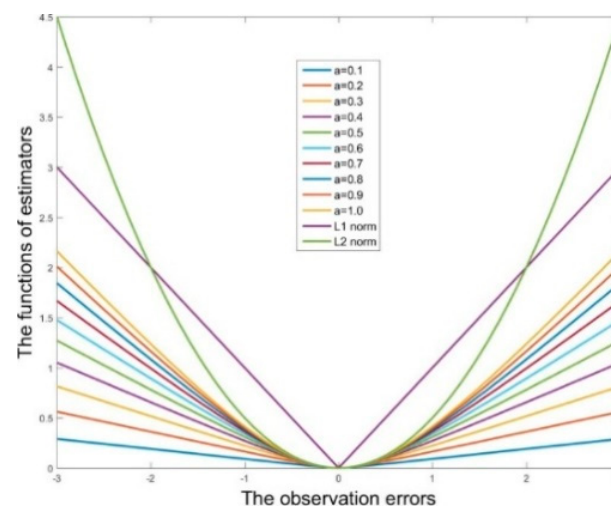


Figure 2. Curve of $\psi(t)$ with different a values.

In $\omega_r(i, j, k, l)$, Equation (8), L₂ norm was used to measure the difference between the point $y(i, j)$ and $y(k, l)$. Theoretically, when the difference is large, the weight of $y(k, l)$ is correspondingly small, which means that $y(k, l)$ has little impact on the estimation of $y(i, j)$. Meanwhile, when the difference is very small, especially close to 0, $y(k, l)$ is similar to $y(i, j)$. Here, the L₂ norm can enlarge the weight of $y(k, l)$, which increases its contribution to $y(i, j)$. Sometimes, the expected effect may not be achieved using the L₂ norm.

The intensity difference in the smooth region tending to be zero is an ideal condition; thus, using the L_2 norm to control the weight based on the intensity similarity is not an optimal choice. A simple example is shown in Figure 3 where the region in the red block is an almost smooth area, which is noted as region A . Assuming that $y(i, j) \in A, y(k, l) \in A, y(i, j)$ is a corrupted pixel, and $y(k, l)$ is uncorrupted. Intuitively, $y(k, l)$ should make a large contribution for estimating $y(i, j)$, namely, $\omega(i, j, k, l) \rightarrow 1$, where $y(i, j)$ is the center value of region $P1$ and $y(k, l)$ is the center value of region $P2$. However, in the real situation, this assumption could not be achieved because the L_2 norm will sharpen the difference. However, the adaptive norm would have a good performance. The analysis is discussed as follows.

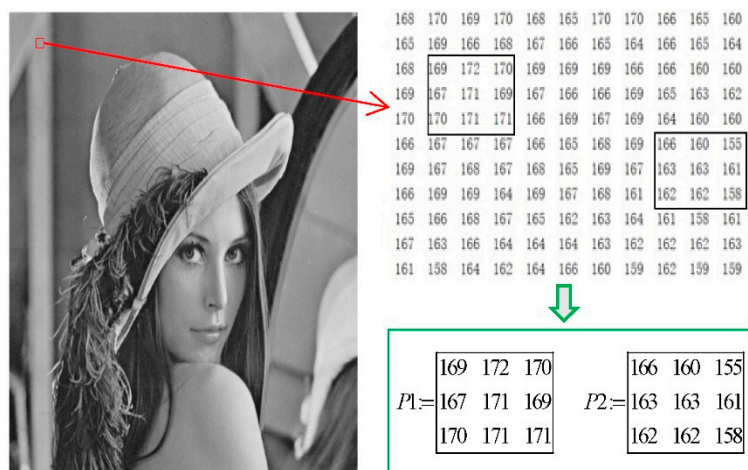


Figure 3. Simple example.

Let $\hat{\omega}_r^{adaptive}$ and $\hat{\omega}_r^{L2}$ denote the Gaussian functions to measure the intensity similarity based on the adaptive norm and L_2 norm. Formally, they can be presented as

$$\hat{\omega}_r^{adaptive}(i, j, k, l) = \exp\left(-\frac{\sum_{\substack{y(p, q) \in P1 \\ y(r, s) \in P2}} \psi(y(p, q) - y(r, s), a_{p, q, r, s})}{\beta}\right) \tag{23}$$

$$\hat{\omega}_r^{L2}(i, j, k, l) = \exp\left(-\frac{\sum_{\substack{y(p, q) \in P1 \\ y(r, s) \in P2}} (y(p, q) - y(r, s))^2}{\beta}\right) \tag{24}$$

From these expressions, the difference measurement part of the two formulas is $(\psi(y(p, q) - y(r, s), a_{p, q, r, s}))$ and $(y(p, q) - y(r, s))^2$. Here, we need to prove that $\psi(y(p, q) - y(r, s), a_{p, q, r, s}) < (y(p, q) - y(r, s))^2$ with the same σ_r . For convenience, denote $y(p, q) - y(r, s) = t$, such that

$$g(t) = a_{p, q, r, s} \sqrt{a_{p, q, r, s}^2 + t^2} - a_{p, q, r, s}^2 - t^2 = \sqrt{a_{p, q, r, s}^2 + t^2} (a_{p, q, r, s} - \sqrt{a_{p, q, r, s}^2 + t^2}) \tag{25}$$

Obviously, $g(t) < 0, \hat{\omega}_r^{adaptive}(i, j, k, l) > \hat{\omega}_r^{L2}(i, j, k, l)$. Specifically, using the L_2 norm to measure the difference will weaken the contribution of $y(k, l)$, which is contrary to the proposed norm.

6. Experimental Results and Discussion

To evaluate the proposed algorithm, several simulation experiments were performed in this section to compare with many state-of-the-art methods. The noisy images are generated by adding salt and pepper noise with nine different intensities from $\sigma = 10\%$ to $\sigma = 90\%$ to the test images shown in Figure 4, which presents gray-scaled image with sizes between 256×256 and 512×512 . Denoising performance was measured by many subjective and objective standards. The subjective standard included reconstructed images and the corresponding error images. The objective standard included image enhancement factor (IEF) [35], structure similarity index (SSIM) [36], and peak signal-to-noise ratio (PSNR) to evaluate the performance of the proposed method.

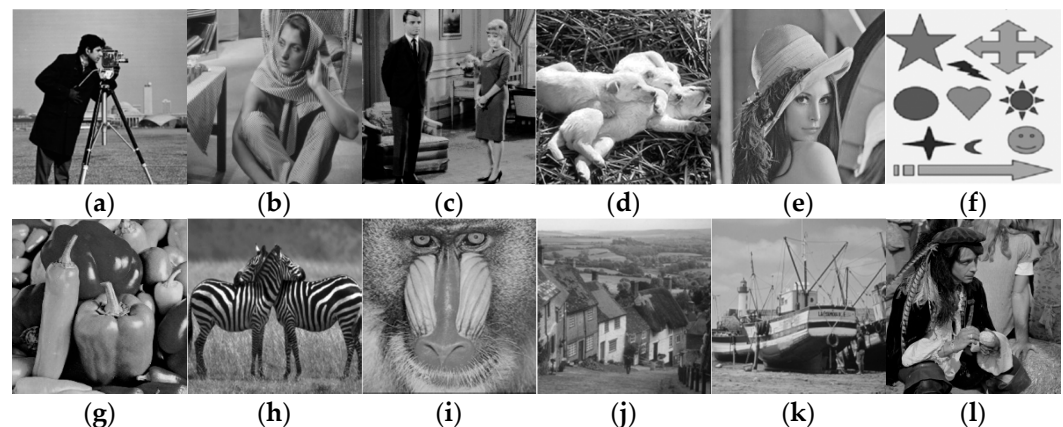


Figure 4. Twelve test images. (a) Cameraman; (b) Barbara; (c) Couple; (d) Dog; (e) Lena; (f) Man-made; (g) Pepper; (h) Zebra; (i) Baboon; (j) Street; (k) Boat; (l) Man.

In this work, we first compared the proposed method using the adaptive norm with L_1 norm- and L_2 norm-based methods. Then, we compared the proposed method with eight previous works, including the median filter (MF), adaptive center weighted median filter (ACWME, 2001) [37], NAFSMF (2010) [16], an edge-preserving approach based on adaptive fuzzy switching median filter (NASEPF, 2011) [38], INLM (2016) [31], different applied median filter (DAMF, 2018) [39], decision-based algorithm (DBA, 2011) [40] and adaptive type-2 fuzzy approach for salt-and-pepper noise (FSAP, 2018) [41].

In the proposed method, the related parameters are set as follows: $T_1 = 10$, $T_2 = 30$, $\alpha = 50 \times \text{noise intensity}$, $\beta = 50 \times \text{noise intensity}$, the similar patch size is 3×3 , and the searching window size is 7×7 . For the other methods, the parameters are set for the best performance. All codes were running in MATLAB 2015b using an i7-7500U CPU, 16 GB RAM computer under Microsoft Windows 10 operation system. For each comparison method, all parameters were selected under the declarations in their papers.

6.1. Comparison under Different Norm Choice

We will take Barbara and Man-made images shown in Figure 4 as examples to compare our proposed method with the L_1 -based method and L_2 -based method. The main difference between these three methods is the norm used to measure the intensity similarity in the second stage.

Tables 1–3 show the PSNR value, SSIM value, and IEF value, respectively. According to the results in Tables 1–3, the proposed method outperformed the other two methods in most cases, especially in high-intensity noise. For example, the proposed method performs better than the L_1 -based method (average PSNR(dB): 0.68, 0.66, and 0.12; average SSIM: 0.0151, 0.0178, and 0.0099; average IEF: 10.93, 5.85, and 1.09) in the case of $\sigma = 70\%$, $\sigma = 80\%$, and $\sigma = 90\%$, respectively. Moreover, the proposed method performs better than the L_2 -based method (average PSNR(dB): 0.69, 1.1, and 1.1; average SSIM: 0.0029, 0.0132, and 0.0172; average IEF: 6.65, 13.74, and 13.14) in the case of $\sigma = 70\%$, $\sigma = 80\%$, and $\sigma = 90\%$,

respectively. The L_1 -based method has the advantage in edge preserving. However, the detail region would be over-smooth. The L_2 -based method can remove noise effectively, while it always makes the edges blur. The proposed method is based on an adaptive norm which has self-adaptability of mimicking L_1 and L_2 norms. Thus, our method can balance the noise removal and detail protection, even in high-intensity noise.

Table 1. PSNR comparison for our proposed method, L_1 -based method, and L_2 -based method with the noise intensity varied from 10% to 90%.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|----------|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Barbara | OURS | 36.28 | 32.93 | 31.07 | 29.60 | 28.64 | 27.42 | 26.38 | 25.21 | 23.47 |
| | L_1 -based | 36.19 | 32.60 | 30.77 | 29.31 | 28.40 | 27.24 | 26.22 | 24.99 | 23.40 |
| | L_2 -based | 36.78 | 33.38 | 30.87 | 29.21 | 27.95 | 26.53 | 25.46 | 24.23 | 22.34 |
| Man-made | OURS | 43.61 | 38.83 | 35.96 | 33.13 | 31.05 | 29.33 | 27.44 | 25.74 | 22.39 |
| | L_1 -based | 42.80 | 38.66 | 35.54 | 32.77 | 31.54 | 29.19 | 26.24 | 24.65 | 22.23 |
| | L_2 -based | 42.34 | 38.48 | 35.16 | 32.28 | 30.59 | 28.22 | 26.99 | 24.46 | 21.30 |

Table 2. SSIM comparison for our proposed method, L_1 -based method, and L_2 -based method with the noise intensity varied from 10% to 90%.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|----------|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Barbara | OURS | 0.9837 | 0.9643 | 0.9431 | 0.9179 | 0.8955 | 0.8641 | 0.8250 | 0.7766 | 0.6927 |
| | L_1 -based | 0.9829 | 0.9617 | 0.9392 | 0.9126 | 0.8898 | 0.8579 | 0.8178 | 0.7718 | 0.6901 |
| | L_2 -based | 0.9863 | 0.9697 | 0.9448 | 0.9180 | 0.8880 | 0.8496 | 0.8079 | 0.7519 | 0.6639 |
| Man-made | OURS | 0.9982 | 0.9945 | 0.9892 | 0.9795 | 0.9682 | 0.9529 | 0.9298 | 0.9018 | 0.8285 |
| | L_1 -based | 0.9978 | 0.9943 | 0.9886 | 0.9796 | 0.9738 | 0.9560 | 0.9069 | 0.8711 | 0.8113 |
| | L_2 -based | 0.9980 | 0.9952 | 0.9900 | 0.9820 | 0.9731 | 0.9549 | 0.9411 | 0.9001 | 0.8229 |

Table 3. IEF comparison for our proposed method, L_1 -based method, and L_2 -based method with the noise intensity varied from 10% to 90%.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|----------|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Barbara | OURS | 140.416 | 130.547 | 127.952 | 122.382 | 122.607 | 110.510 | 101.752 | 88.895 | 67.242 |
| | L_1 -based | 139.339 | 121.283 | 120.390 | 114.770 | 116.538 | 106.415 | 98.443 | 85.115 | 66.178 |
| | L_2 -based | 160.801 | 146.468 | 123.910 | 113.409 | 105.971 | 91.0615 | 83.418 | 71.694 | 52.399 |
| Man-made | OURS | 631.958 | 582.068 | 443.847 | 341.676 | 299.842 | 208.344 | 131.921 | 97.475 | 58.901 |
| | L_1 -based | 753.249 | 562.153 | 421.249 | 295.480 | 281.036 | 192.859 | 113.374 | 89.552 | 57.791 |
| | L_2 -based | 678.310 | 542.347 | 388.393 | 265.755 | 226.931 | 154.95 | 136.959 | 87.188 | 47.458 |

6.2. Comparisons between Pre- and Post-Processed Images

In this sub-section, we would like to show the proposed method effectively reducing the error for a preprocessed image. Figure 5 shows three examples based on Couple, Pepper, and Street images. For better comparisons, we show the error images and the enlarged detail parts. From Figure 5, we can see that the post-processed images have more acceptable details and less error. Besides, the corresponding PSNR, SSIM, and IEF values are presented in Tables 4–6.

According to these tales, it can be seen that the post-processed images achieves higher PSNR, SSIM, and IEF values. For example, the post-processed images gains higher values than the pre-processed images (average PSNR(dB): 1.41, 1.68, and 2.72; average SSIM: 0.0320, 0.0432, and 0.1031; average IEF: 36.47, 33.46, and 37.94) in the case of $\sigma = 70\%$, $\sigma = 80\%$, and $\sigma = 90\%$, respectively. This indicates that the proposed method could further reduce the error in the preprocessed images. That is because in the proposed two-stage method a novel bilateral filter is used to weaken the median-type filter errors is effective.

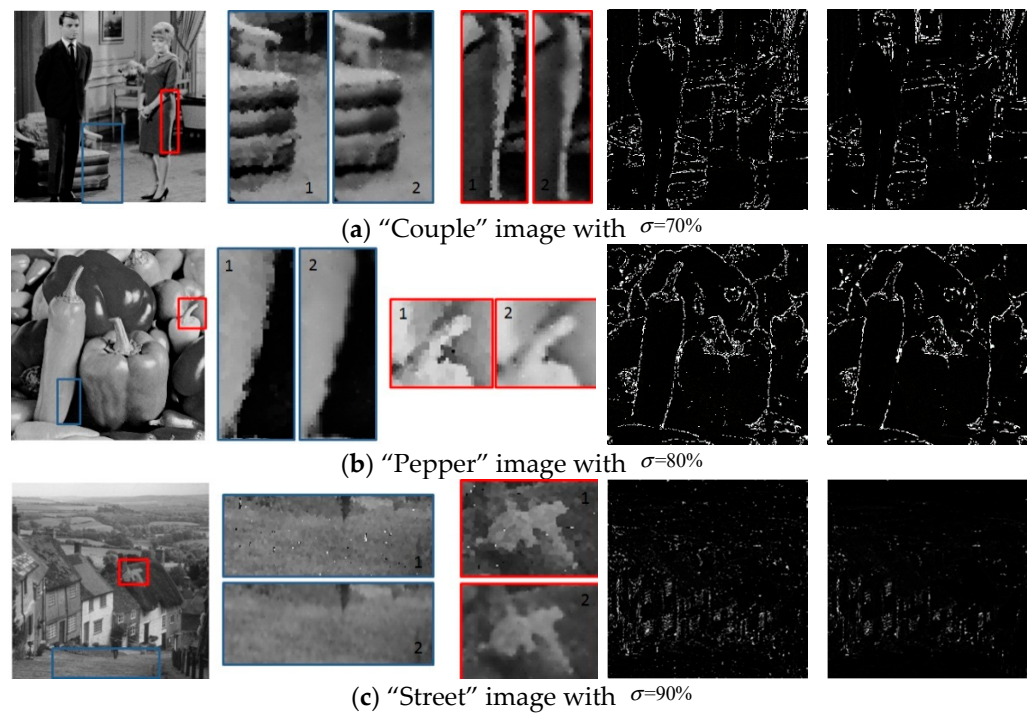


Figure 5. Comparisons of the pre- and post-processed images. In each line, the left part is the original image, the middle part is the zoomed details. The pre-processed marked "1", the post-processed marked "2". The right part is the corresponding error images, the first one is pre-processed error, the second one is post-processed error.

Table 4. PSNR Comparisons between pre- and post-processed images.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|--------|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Couple | Pre-processed | 38.10 | 34.08 | 31.48 | 29.68 | 28.07 | 26.53 | 25.27 | 23.52 | 20.48 |
| | Post-processed | 39.10 | 35.29 | 32.73 | 30.86 | 29.21 | 27.79 | 26.64 | 25.06 | 22.86 |
| Pepper | Pre-processed | 37.54 | 33.87 | 31.50 | 29.61 | 28.05 | 26.44 | 24.78 | 23.14 | 20.14 |
| | Post-processed | 37.76 | 35.01 | 32.91 | 31.15 | 29.62 | 28.01 | 26.54 | 25.13 | 22.65 |
| Street | Pre-processed | 39.36 | 35.82 | 33.47 | 31.95 | 30.50 | 29.24 | 27.80 | 26.25 | 22.57 |
| | Post-processed | 40.24 | 36.40 | 34.07 | 32.48 | 31.14 | 30.08 | 28.90 | 27.76 | 25.84 |

Table 5. SSIM Comparisons between pre- and post-processed images.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|--------|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Couple | Pre-processed | 0.9859 | 0.9706 | 0.9500 | 0.9280 | 0.9026 | 0.8662 | 0.8269 | 0.7626 | 0.6148 |
| | Post-processed | 0.9893 | 0.9764 | 0.9580 | 0.9368 | 0.9125 | 0.8803 | 0.8476 | 0.7789 | 0.6973 |
| Pepper | Pre-processed | 0.9871 | 0.9707 | 0.9547 | 0.9300 | 0.9052 | 0.8727 | 0.8310 | 0.7735 | 0.6456 |
| | Post-processed | 0.9883 | 0.9749 | 0.9617 | 0.9410 | 0.9221 | 0.8966 | 0.8687 | 0.8300 | 0.7590 |
| Street | Pre-processed | 0.9871 | 0.9707 | 0.9547 | 0.9300 | 0.9052 | 0.8727 | 0.8310 | 0.7735 | 0.6456 |
| | Post-processed | 0.9883 | 0.9749 | 0.9617 | 0.9410 | 0.9221 | 0.8966 | 0.8687 | 0.8300 | 0.7590 |

Table 6. IEF Comparisons between pre- and post-processed images.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|--------|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Couple | Pre-processed | 232.16 | 180.32 | 148.58 | 130.69 | 112.86 | 95.43 | 83.98 | 62.35 | 35.61 |
| | Post-processed | 284.77 | 226.29 | 194.08 | 168.83 | 145.19 | 126.20 | 113.82 | 80.40 | 60.58 |
| Pepper | Pre-processed | 221.37 | 197.51 | 163.91 | 141.82 | 122.67 | 102.06 | 81.17 | 63.20 | 35.82 |
| | Post-processed | 248.20 | 248.25 | 222.90 | 199.28 | 174.44 | 145.03 | 120.45 | 98.71 | 63.11 |
| Street | Pre-processed | 295.41 | 260.04 | 229.59 | 215.14 | 191.47 | 172.26 | 144.31 | 115.03 | 55.76 |
| | Post-processed | 345.85 | 293.34 | 261.28 | 241.07 | 220.54 | 207.64 | 184.61 | 161.84 | 117.33 |

6.3. Subjective Quality Analysis

In this sub-section, three evaluation measures (i.e., PSNR, IEF, and SSIM) were presented to validate the perception-based image quality assessment on different images under different noise levels. We compared the proposed adaptive method with DBA, MF, ACWMF, NASFM, NASEPF, INLM, DAMF, and FSAP. For fair comparison, all parameters in each algorithm are under their statements in published papers.

The graphs of IEF for different images with noise levels varying from $\sigma = 10\text{--}90\%$ are shown in Figure 6. For the subfigures in Figure 6, x -axis means $\sigma = 10\text{--}90\%$ from left to right. The y -axis denotes the IEF value. This figure confirmed that the proposed method obtained better results compared with other considered methods in terms of higher IEF value. Table 7 shows the PSNR values on the restored images corresponding to eight test images, including those shown in Figure 4 with the noise intensity changes from 10% to 90% as well as Table 8 shows the corresponding SSIM values on the restored images. For better comparisons, we have colored the best values into red.

From these tables, we can see that our proposed method gets the highest PSNR and SSIM values both in the low noise intensity and high noise intensity, which demonstrates the efficiency and necessity of incorporating a novel bilateral filter to weaken the median-type filter errors. In the case of low noise intensity, taking $\sigma = 10\%$ as an example, all the methods achieve high PSNR and SSIM values. The PSNR and SSIM values achieved by different methods are shown in the following: MF (22.01 dB, 0.7025), ACWMF (31.20 dB, 0.9524), DBA (35.09 dB, 0.9828), NAFSM (36.71 dB, 0.9807), NASEPF (27.25 dB, 0.8077), INLM (30.52 dB, 0.8750), DAMF (36.93 dB, 0.9850), FSAP (35.56 dB, 0.9831), and OURS (38.15 dB, 0.9865). We can see that in the case of $\sigma = 10\%$, the PSNR value of the proposed method (38.15 dB) has outperformed that of the MF method, the ACWMF method, the DBA method, the DAMF method, and the FSAP method more than 1.2. Even in the case of $\sigma = 90\%$ (correspondingly a strong noise), the PSNR value of the proposed method (22.81 dB) has outperformed the PSNR values of the MF method, the ACWMF method, the DBA method, the DAMF method, and the FSAP method more than 1.4. In addition, the proposed method also achieves slightly better reconstruction performance than the INLM method (22.27 dB). Compared with the INLM, a novel bilateral filter is proposed in this work, which is more efficient to suppress the Gaussian–Laplacian errors.

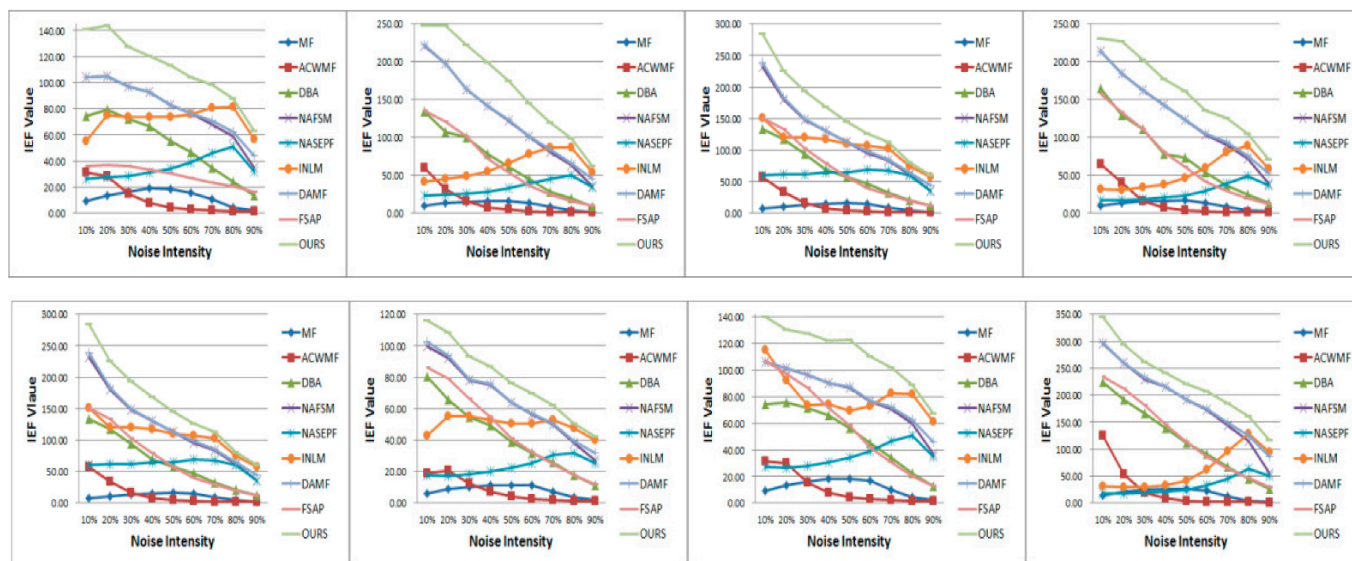


Figure 6. Comparison of different methods. From top left to right down: the curves of IEF vs. noise intensity achieved for the images Baboon, Pepper, Man-made, Lena, Couple, Cameraman, Barbara, Street.

Table 7. Comparison of PSNR values for test images under different noise intensities.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|-----------|--------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Barbara | MF | 24.13 | 22.74 | 21.88 | 21.17 | 20.28 | 19.18 | 16.19 | 11.88 | 7.69 |
| | ACWMF | 29.79 | 26.60 | 21.95 | 17.84 | 14.21 | 11.53 | 9.17 | 7.41 | 5.83 |
| | DBA | 33.56 | 30.69 | 28.68 | 27.07 | 25.45 | 23.72 | 21.89 | 19.54 | 16.50 |
| | NAFSM | 34.97 | 31.78 | 29.78 | 28.21 | 27.10 | 25.77 | 24.73 | 23.44 | 20.82 |
| | NASEPF | 29.28 | 26.14 | 24.56 | 23.69 | 23.17 | 22.89 | 22.94 | 22.72 | 20.55 |
| | INLM | 35.41 | 31.47 | 28.70 | 27.48 | 26.23 | 25.67 | 25.51 | 24.86 | 23.07 |
| | DAMF | 34.97 | 31.78 | 29.78 | 28.21 | 27.12 | 25.80 | 24.82 | 23.64 | 21.79 |
| | FSAP | 35.04 | 31.64 | 29.35 | 27.26 | 25.40 | 23.23 | 21.20 | 18.87 | 16.49 |
| | OURS | 36.28 | 32.93 | 31.07 | 29.60 | 28.64 | 27.42 | 26.38 | 25.21 | 23.47 |
| Baboon | MF | 19.58 | 19.32 | 18.98 | 18.65 | 18.26 | 17.34 | 15.27 | 11.31 | 7.48 |
| | ACWMF | 25.76 | 23.45 | 20.16 | 16.81 | 13.75 | 11.14 | 9.02 | 7.24 | 5.80 |
| | DBA | 28.02 | 25.96 | 24.24 | 22.84 | 21.59 | 20.24 | 19.04 | 17.74 | 16.15 |
| | NAFSM | 30.87 | 27.80 | 25.90 | 24.35 | 23.05 | 21.88 | 20.72 | 19.49 | 17.69 |
| | NASEPF | 26.56 | 23.62 | 22.03 | 21.01 | 20.36 | 19.94 | 19.57 | 19.05 | 17.55 |
| | INLM | 28.93 | 26.79 | 25.06 | 23.56 | 22.49 | 21.73 | 21.08 | 20.38 | 19.20 |
| | DAMF | 30.87 | 27.80 | 25.90 | 24.35 | 23.05 | 21.89 | 20.75 | 19.58 | 18.15 |
| | FSAP | 30.35 | 27.45 | 25.60 | 24.03 | 22.62 | 21.37 | 20.13 | 18.97 | 17.49 |
| | OURS | 31.23 | 28.04 | 26.08 | 24.53 | 23.33 | 22.35 | 21.41 | 20.49 | 19.35 |
| Cameraman | MF | 21.95 | 21.10 | 20.11 | 19.18 | 18.43 | 17.53 | 15.17 | 11.42 | 7.58 |
| | ACWMF | 27.76 | 25.24 | 21.31 | 17.63 | 14.38 | 11.60 | 9.55 | 7.59 | 6.09 |
| | DBA | 34.17 | 30.40 | 27.86 | 26.04 | 24.16 | 22.44 | 20.97 | 18.74 | 16.34 |
| | NAFSM | 35.01 | 31.73 | 29.24 | 27.69 | 26.12 | 24.75 | 23.59 | 21.82 | 19.73 |
| | NASEPF | 27.70 | 24.69 | 23.11 | 22.16 | 21.68 | 21.43 | 21.56 | 21.01 | 19.48 |
| | INLM | 31.51 | 29.66 | 27.85 | 26.31 | 25.23 | 24.42 | 23.94 | 22.86 | 21.54 |
| | DAMF | 35.13 | 31.80 | 29.27 | 27.71 | 26.13 | 24.78 | 23.67 | 21.94 | 20.46 |
| | FSAP | 34.42 | 31.14 | 28.58 | 26.35 | 24.25 | 22.45 | 20.69 | 18.67 | 16.43 |
| | OURS | 37.71 | 32.62 | 30.07 | 28.49 | 26.93 | 25.74 | 24.62 | 23.09 | 21.78 |
| Couple | MF | 22.79 | 21.75 | 21.06 | 20.24 | 19.66 | 18.45 | 15.48 | 11.52 | 7.56 |
| | ACWMF | 32.10 | 26.98 | 22.15 | 17.65 | 14.20 | 11.29 | 8.90 | 7.15 | 5.69 |
| | DBA | 35.84 | 32.34 | 29.68 | 27.11 | 25.38 | 23.64 | 21.34 | 19.22 | 16.13 |
| | NAFSM | 38.10 | 34.08 | 31.48 | 29.68 | 28.07 | 26.53 | 25.27 | 23.52 | 20.48 |
| | NASEPF | 32.34 | 29.45 | 27.73 | 26.65 | 25.73 | 25.18 | 24.40 | 23.41 | 20.49 |
| | INLM | 36.28 | 32.37 | 30.62 | 29.29 | 28.02 | 27.07 | 26.18 | 24.41 | 22.66 |
| | DAMF | 38.22 | 34.14 | 31.51 | 29.70 | 28.09 | 26.65 | 25.32 | 23.81 | 21.49 |
| | FSAP | 36.27 | 32.80 | 29.92 | 27.53 | 25.19 | 22.80 | 20.86 | 18.59 | 15.87 |
| | OURS | 39.10 | 35.29 | 32.73 | 30.86 | 29.21 | 27.79 | 26.64 | 25.06 | 22.86 |
| Lena | MF | 25.00 | 23.39 | 22.32 | 21.15 | 20.49 | 18.81 | 16.26 | 12.11 | 8.20 |
| | ACWMF | 33.45 | 28.29 | 22.53 | 18.26 | 14.79 | 11.94 | 9.68 | 7.76 | 6.35 |
| | DBA | 37.54 | 33.34 | 30.96 | 28.24 | 27.02 | 24.89 | 22.64 | 20.33 | 17.48 |
| | NAFSM | 38.54 | 34.73 | 32.46 | 30.66 | 29.10 | 27.56 | 26.31 | 24.71 | 21.55 |
| | NASEPF | 27.99 | 24.84 | 23.40 | 22.47 | 22.13 | 22.17 | 22.72 | 23.03 | 21.25 |
| | INLM | 30.43 | 27.24 | 25.85 | 25.09 | 25.07 | 25.30 | 25.90 | 25.68 | 23.39 |
| | DAMF | 38.54 | 34.73 | 32.46 | 30.66 | 29.11 | 27.58 | 26.46 | 24.95 | 22.82 |
| | FSAP | 37.25 | 33.37 | 30.84 | 28.23 | 26.08 | 23.62 | 21.62 | 19.27 | 17.00 |
| | OURS | 39.25 | 35.73 | 33.46 | 31.63 | 30.30 | 28.76 | 27.77 | 26.36 | 24.15 |
| Peppers | MF | 23.86 | 22.26 | 21.08 | 19.92 | 19.18 | 17.84 | 14.80 | 10.95 | 6.90 |
| | ACWMF | 31.92 | 26.11 | 21.58 | 16.91 | 13.93 | 10.87 | 8.56 | 6.80 | 5.32 |
| | DBA | 35.49 | 31.38 | 29.51 | 27.21 | 25.25 | 23.15 | 20.54 | 18.52 | 14.67 |
| | NAFSM | 37.54 | 33.87 | 31.50 | 29.61 | 28.05 | 26.44 | 24.78 | 23.14 | 20.14 |
| | NASEPF | 28.10 | 25.12 | 23.73 | 22.75 | 22.42 | 22.32 | 22.29 | 22.19 | 19.96 |
| | INLM | 30.49 | 27.67 | 26.42 | 25.66 | 25.49 | 25.38 | 25.14 | 24.59 | 21.97 |
| | DAMF | 37.55 | 33.89 | 31.51 | 29.62 | 28.07 | 26.45 | 24.93 | 23.36 | 21.17 |
| | FSAP | 35.43 | 31.79 | 29.47 | 26.77 | 24.52 | 22.19 | 19.68 | 17.31 | 14.79 |
| | OURS | 37.76 | 35.01 | 32.91 | 31.15 | 29.62 | 28.01 | 26.54 | 25.13 | 22.65 |

Table 7. Cont.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|----------|--------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Street | MF | 26.12 | 24.94 | 23.64 | 22.73 | 21.88 | 20.45 | 17.01 | 12.06 | 7.60 |
| | ACWMF | 35.74 | 29.04 | 22.77 | 18.02 | 14.43 | 11.46 | 9.19 | 7.36 | 5.82 |
| | DBA | 38.26 | 34.61 | 32.19 | 30.20 | 28.33 | 26.62 | 24.64 | 22.34 | 19.47 |
| | NAFSM | 39.36 | 35.82 | 33.47 | 31.95 | 30.50 | 29.24 | 27.80 | 26.25 | 22.57 |
| | NASEPF | 27.34 | 24.42 | 22.88 | 22.11 | 21.84 | 22.12 | 22.86 | 23.74 | 22.13 |
| | INLM | 29.67 | 26.62 | 24.88 | 24.08 | 23.98 | 24.89 | 26.12 | 26.82 | 24.88 |
| | DAMF | 39.38 | 35.83 | 33.51 | 31.96 | 30.51 | 29.29 | 27.93 | 26.58 | 24.42 |
| | FSAP | 38.36 | 34.95 | 32.47 | 30.27 | 28.23 | 26.29 | 24.32 | 22.36 | 19.87 |
| | OURS | 40.24 | 36.40 | 34.07 | 32.48 | 31.14 | 30.08 | 28.90 | 27.76 | 25.84 |
| Man-made | MF | 23.59 | 20.57 | 18.51 | 17.21 | 16.30 | 15.02 | 13.10 | 10.19 | 6.47 |
| | ACWMF | 33.04 | 26.93 | 21.64 | 17.32 | 13.91 | 10.74 | 8.63 | 6.81 | 5.22 |
| | DBA | 37.81 | 33.02 | 30.40 | 27.43 | 24.65 | 22.75 | 20.09 | 17.59 | 14.38 |
| | NAFSM | 39.29 | 35.26 | 32.83 | 30.54 | 28.89 | 27.35 | 25.40 | 23.69 | 19.63 |
| | NASEPF | 18.69 | 15.76 | 14.24 | 13.47 | 13.35 | 13.83 | 15.01 | 16.77 | 17.98 |
| | INLM | 21.43 | 18.27 | 17.10 | 16.99 | 17.63 | 18.83 | 20.82 | 22.60 | 21.48 |
| | DAMF | 40.74 | 35.91 | 33.19 | 30.72 | 28.98 | 27.44 | 25.44 | 23.95 | 20.72 |
| | FSAP | 37.35 | 33.26 | 30.59 | 27.86 | 24.98 | 23.11 | 20.56 | 18.13 | 15.41 |
| | OURS | 43.61 | 38.83 | 35.96 | 33.13 | 31.05 | 29.33 | 27.44 | 25.74 | 22.39 |

Table 8. Comparison of SSIM values for test images under different noise intensities.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|-----------|--------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Barbara | MF | 0.7048 | 0.6969 | 0.6852 | 0.6749 | 0.6471 | 0.6084 | 0.4466 | 0.2035 | 0.0466 |
| | ACWMF | 0.9494 | 0.8918 | 0.7373 | 0.5042 | 0.2947 | 0.1513 | 0.0785 | 0.0377 | 0.0155 |
| | DBA | 0.9769 | 0.9507 | 0.9174 | 0.8797 | 0.8319 | 0.7729 | 0.6864 | 0.5678 | 0.3945 |
| | NAFSM | 0.9790 | 0.9562 | 0.9288 | 0.8978 | 0.8659 | 0.8243 | 0.7781 | 0.7144 | 0.5882 |
| | NASEPF | 0.8877 | 0.8280 | 0.7883 | 0.7539 | 0.7224 | 0.6946 | 0.6665 | 0.6379 | 0.5625 |
| | INLM | 0.9701 | 0.9319 | 0.8837 | 0.8508 | 0.8184 | 0.7969 | 0.7794 | 0.7523 | 0.6743 |
| | DAMF | 0.9790 | 0.9562 | 0.9288 | 0.8978 | 0.8662 | 0.8251 | 0.7809 | 0.7207 | 0.6285 |
| | FSAP | 0.9798 | 0.9578 | 0.9265 | 0.8836 | 0.8289 | 0.7484 | 0.6433 | 0.5007 | 0.3509 |
| | OURS | 0.9834 | 0.9641 | 0.9427 | 0.9174 | 0.8949 | 0.8633 | 0.8242 | 0.7757 | 0.6898 |
| Baboon | MF | 0.3842 | 0.3813 | 0.3777 | 0.3726 | 0.3630 | 0.3336 | 0.2545 | 0.1092 | 0.0253 |
| | ACWMF | 0.9189 | 0.8578 | 0.7191 | 0.5211 | 0.3162 | 0.1706 | 0.0857 | 0.0412 | 0.0188 |
| | DBA | 0.9673 | 0.9287 | 0.8802 | 0.8234 | 0.7536 | 0.6681 | 0.5676 | 0.4500 | 0.3101 |
| | NAFSM | 0.9704 | 0.9373 | 0.8989 | 0.8557 | 0.8031 | 0.7428 | 0.6688 | 0.5745 | 0.4240 |
| | NASEPF | 0.8633 | 0.7904 | 0.7342 | 0.6867 | 0.6388 | 0.5933 | 0.5462 | 0.4939 | 0.3964 |
| | INLM | 0.9141 | 0.8785 | 0.8306 | 0.7759 | 0.7224 | 0.6733 | 0.6225 | 0.5618 | 0.4605 |
| | DAMF | 0.9704 | 0.9373 | 0.8989 | 0.8557 | 0.8032 | 0.7432 | 0.6700 | 0.5789 | 0.4463 |
| | FSAP | 0.9683 | 0.9345 | 0.8929 | 0.8407 | 0.7700 | 0.6868 | 0.5809 | 0.4632 | 0.3315 |
| | OURS | 0.9711 | 0.9367 | 0.8956 | 0.8490 | 0.7949 | 0.7362 | 0.6666 | 0.5857 | 0.4724 |
| Cameraman | MF | 0.7256 | 0.7206 | 0.7120 | 0.6988 | 0.6788 | 0.6358 | 0.4842 | 0.1839 | 0.0360 |
| | ACWMF | 0.9398 | 0.8774 | 0.7122 | 0.4772 | 0.2626 | 0.1323 | 0.0790 | 0.0398 | 0.0207 |
| | DBA | 0.9840 | 0.9645 | 0.9373 | 0.9075 | 0.8676 | 0.8140 | 0.7649 | 0.6753 | 0.5723 |
| | NAFSM | 0.9796 | 0.9629 | 0.9422 | 0.9228 | 0.8957 | 0.8636 | 0.8232 | 0.7633 | 0.6456 |
| | NASEPF | 0.7435 | 0.6641 | 0.6221 | 0.5928 | 0.5697 | 0.5501 | 0.5431 | 0.5408 | 0.5597 |
| | INLM | 0.8752 | 0.8514 | 0.8054 | 0.7627 | 0.7395 | 0.7315 | 0.7462 | 0.7450 | 0.7125 |
| | DAMF | 0.9862 | 0.9705 | 0.9493 | 0.9277 | 0.8998 | 0.8673 | 0.8300 | 0.7727 | 0.7039 |
| | FSAP | 0.9844 | 0.9668 | 0.9423 | 0.9107 | 0.8671 | 0.8089 | 0.7411 | 0.6466 | 0.5256 |
| | OURS | 0.9870 | 0.9717 | 0.9513 | 0.9295 | 0.9035 | 0.8746 | 0.8423 | 0.7933 | 0.7362 |

Table 8. Cont.

| Image | Method | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|----------|--------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Couple | MF | 0.6941 | 0.6862 | 0.6758 | 0.6568 | 0.6365 | 0.5853 | 0.4275 | 0.2018 | 0.0521 |
| | ACWMF | 0.9633 | 0.9022 | 0.7493 | 0.5330 | 0.3277 | 0.1694 | 0.0839 | 0.0420 | 0.0208 |
| | DBA | 0.9861 | 0.9670 | 0.9378 | 0.8996 | 0.8568 | 0.7893 | 0.7024 | 0.5902 | 0.4184 |
| | NAFSM | 0.9859 | 0.9706 | 0.9500 | 0.9280 | 0.9026 | 0.8662 | 0.8269 | 0.7626 | 0.6148 |
| | NASEPF | 0.9397 | 0.9027 | 0.8697 | 0.8454 | 0.8192 | 0.7942 | 0.7652 | 0.7285 | 0.6169 |
| | INLM | 0.9764 | 0.9500 | 0.9272 | 0.9043 | 0.8790 | 0.8514 | 0.8261 | 0.7616 | 0.6809 |
| | DAMF | 0.9887 | 0.9740 | 0.9532 | 0.9311 | 0.9047 | 0.8686 | 0.8289 | 0.7696 | 0.6583 |
| | FSAP | 0.9856 | 0.9686 | 0.9415 | 0.9023 | 0.8443 | 0.7588 | 0.6510 | 0.5156 | 0.3524 |
| | OURS | 0.9893 | 0.9764 | 0.9580 | 0.9368 | 0.9125 | 0.8803 | 0.8476 | 0.7789 | 0.6973 |
| Lena | MF | 0.7651 | 0.7565 | 0.7480 | 0.7332 | 0.7092 | 0.6514 | 0.4768 | 0.1922 | 0.0400 |
| | ACWMF | 0.9733 | 0.9172 | 0.7477 | 0.5051 | 0.2708 | 0.1411 | 0.0691 | 0.0365 | 0.0139 |
| | DBA | 0.9856 | 0.9653 | 0.9416 | 0.9076 | 0.8692 | 0.8118 | 0.7326 | 0.6272 | 0.4832 |
| | NAFSM | 0.9872 | 0.9716 | 0.9540 | 0.9314 | 0.9070 | 0.8728 | 0.8301 | 0.7763 | 0.6457 |
| | NASEPF | 0.8310 | 0.7604 | 0.7202 | 0.6887 | 0.6689 | 0.6504 | 0.6437 | 0.6418 | 0.6035 |
| | INLM | 0.8871 | 0.8441 | 0.8183 | 0.7926 | 0.7817 | 0.7777 | 0.7852 | 0.7834 | 0.7247 |
| | DAMF | 0.9872 | 0.9716 | 0.9540 | 0.9314 | 0.9070 | 0.8734 | 0.8346 | 0.7848 | 0.6918 |
| | FSAP | 0.9857 | 0.9677 | 0.9428 | 0.9039 | 0.8501 | 0.7676 | 0.6655 | 0.5319 | 0.3946 |
| | OURS | 0.9887 | 0.9747 | 0.9588 | 0.9381 | 0.9176 | 0.8898 | 0.8596 | 0.8205 | 0.7461 |
| Peppers | MF | 0.7881 | 0.7765 | 0.7668 | 0.7469 | 0.7237 | 0.6674 | 0.4788 | 0.2234 | 0.0546 |
| | ACWMF | 0.9694 | 0.8945 | 0.7453 | 0.4967 | 0.3105 | 0.1609 | 0.0796 | 0.0417 | 0.0220 |
| | DBA | 0.9856 | 0.9633 | 0.9425 | 0.9058 | 0.8646 | 0.8064 | 0.7134 | 0.6150 | 0.4374 |
| | NAFSM | 0.9871 | 0.9707 | 0.9547 | 0.9300 | 0.9052 | 0.8727 | 0.8310 | 0.7735 | 0.6456 |
| | NASEPF | 0.8726 | 0.8078 | 0.7759 | 0.7430 | 0.7219 | 0.7032 | 0.6881 | 0.6797 | 0.6231 |
| | INLM | 0.9182 | 0.8799 | 0.8609 | 0.8361 | 0.8252 | 0.8194 | 0.8162 | 0.8069 | 0.7411 |
| | DAMF | 0.9874 | 0.9712 | 0.9553 | 0.9306 | 0.9061 | 0.8733 | 0.8347 | 0.7804 | 0.6917 |
| | FSAP | 0.9846 | 0.9650 | 0.9422 | 0.9024 | 0.8498 | 0.7770 | 0.6730 | 0.5464 | 0.3925 |
| | OURS | 0.9883 | 0.9749 | 0.9617 | 0.9410 | 0.9221 | 0.8966 | 0.8687 | 0.8300 | 0.7590 |
| Street | MF | 0.6735 | 0.6683 | 0.6592 | 0.6493 | 0.6303 | 0.5877 | 0.4315 | 0.1796 | 0.0327 |
| | ACWMF | 0.9712 | 0.8991 | 0.7287 | 0.4786 | 0.2635 | 0.1204 | 0.0571 | 0.0266 | 0.0112 |
| | DBA | 0.9818 | 0.9581 | 0.9284 | 0.8899 | 0.8429 | 0.7831 | 0.6988 | 0.5921 | 0.4433 |
| | NAFSM | 0.9837 | 0.9645 | 0.9419 | 0.9160 | 0.8855 | 0.8504 | 0.8024 | 0.7379 | 0.5966 |
| | NASEPF | 0.8713 | 0.8255 | 0.7931 | 0.7644 | 0.7375 | 0.7118 | 0.6820 | 0.6471 | 0.5626 |
| | INLM | 0.9024 | 0.8651 | 0.8364 | 0.8066 | 0.7818 | 0.7625 | 0.7456 | 0.7291 | 0.6696 |
| | DAMF | 0.9841 | 0.9649 | 0.9422 | 0.9164 | 0.8860 | 0.8511 | 0.8047 | 0.7456 | 0.6431 |
| | FSAP | 0.9827 | 0.9628 | 0.9369 | 0.9014 | 0.8550 | 0.7940 | 0.7133 | 0.6174 | 0.4985 |
| | OURS | 0.9858 | 0.9663 | 0.9435 | 0.9173 | 0.8888 | 0.8581 | 0.8190 | 0.7726 | 0.6944 |
| Man-made | MF | 0.8845 | 0.8760 | 0.8615 | 0.8418 | 0.8189 | 0.7423 | 0.5703 | 0.3017 | 0.0752 |
| | ACWMF | 0.9338 | 0.8529 | 0.7048 | 0.5012 | 0.3218 | 0.1728 | 0.0989 | 0.0532 | 0.0222 |
| | DBA | 0.9948 | 0.9850 | 0.9726 | 0.9495 | 0.9181 | 0.8744 | 0.8017 | 0.7175 | 0.5581 |
| | NAFSM | 0.9727 | 0.9625 | 0.9564 | 0.9483 | 0.9438 | 0.9306 | 0.9080 | 0.8721 | 0.7508 |
| | NASEPF | 0.4527 | 0.3886 | 0.3575 | 0.3381 | 0.3294 | 0.3285 | 0.3386 | 0.3679 | 0.4477 |
| | INLM | 0.5564 | 0.4712 | 0.4417 | 0.4281 | 0.4345 | 0.4599 | 0.5299 | 0.6327 | 0.7354 |
| | DAMF | 0.9968 | 0.9913 | 0.9844 | 0.9723 | 0.9606 | 0.9435 | 0.9162 | 0.8847 | 0.8002 |
| | FSAP | 0.9939 | 0.9851 | 0.9724 | 0.9471 | 0.9068 | 0.8615 | 0.7789 | 0.6558 | 0.4802 |
| | OURS | 0.9982 | 0.9945 | 0.9892 | 0.9795 | 0.9682 | 0.9529 | 0.9298 | 0.9018 | 0.8285 |

6.4. Objective Quality Analysis

This section presents the visual results obtained by different methods. We evaluated the performance of the proposed method and the methods under comparison via reconstructed images and the corresponding error images, which showed that the error image presented the difference between the original image and the reconstructed image. In the error image, the white point was the error between the de-noised image and the original image. The whiter point denotes the larger difference.

The subjective analysis of the proposed method against the existing methods with different noise intensities were shown in Figures 7–10. In these figures, the proposed

method was performed in different texture types of images under $\sigma = 60\%$, 70% , 80% , and 90% .

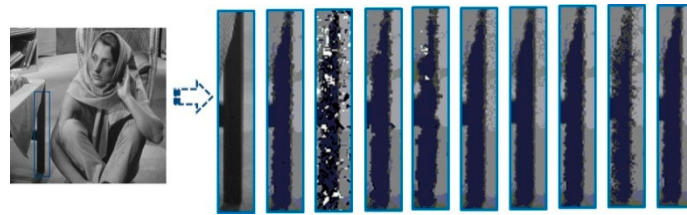


Figure 7. Results of different algorithms for Barbara image with 60% noise intensity. The subfigures in rounded rectangle are corresponding to original image, MF, ACWMF, DBA, NAFSM, NASEPF, INLM, DAMF, FSAP, and our proposed method, respectively.

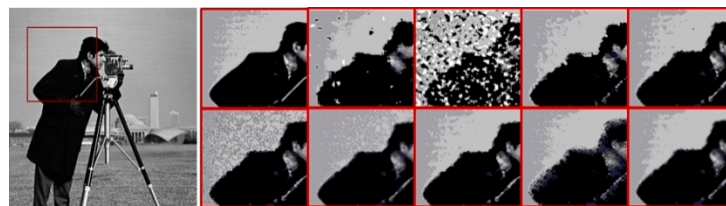


Figure 8. Results of different algorithms for Cameraman image with 70% noise intensity. The subfigures in the first line from left to right are corresponding to original image, MF, ACWMF, DBA, and NAFSM, respectively. The subfigures in the third line from left to right are NASEPF, INLM, DAMF, FSAP, and our proposed method, respectively.

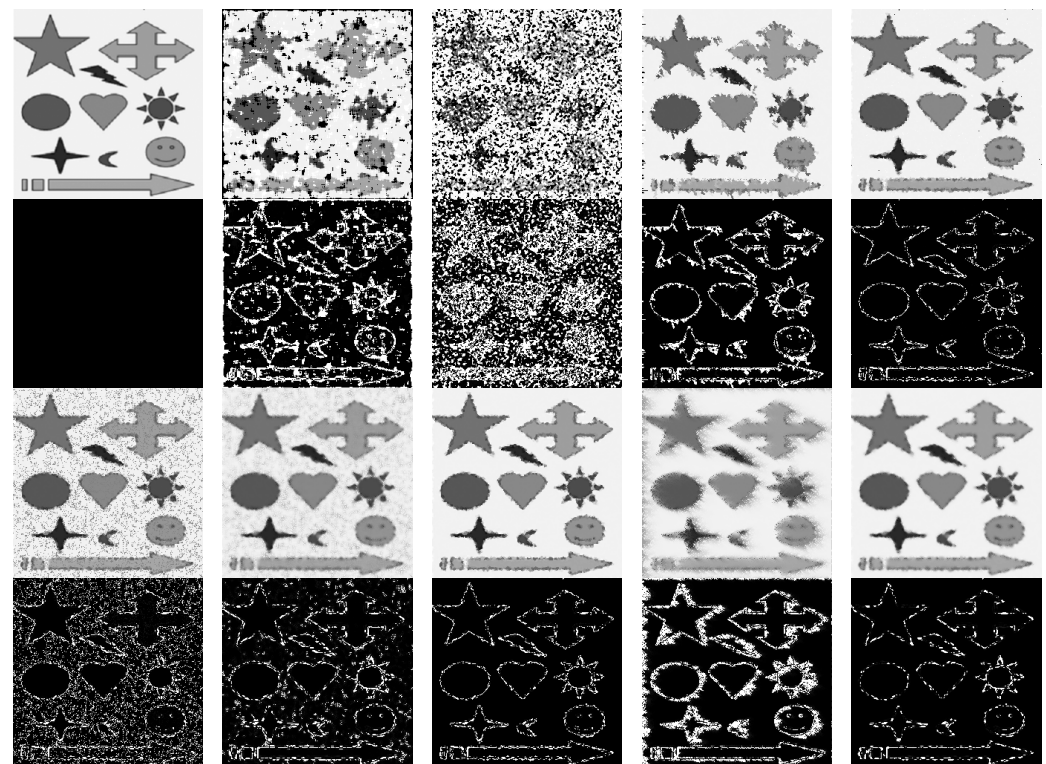


Figure 9. Results of different algorithms for Man-made image with 80% noise intensity. The first and third lines are original image, MF, ACWMF, DBA, NAFSM, NASEPF, INLM, DAMF, FSAP, and our proposed method. The second and fourth lines are noise image and corresponding error images.

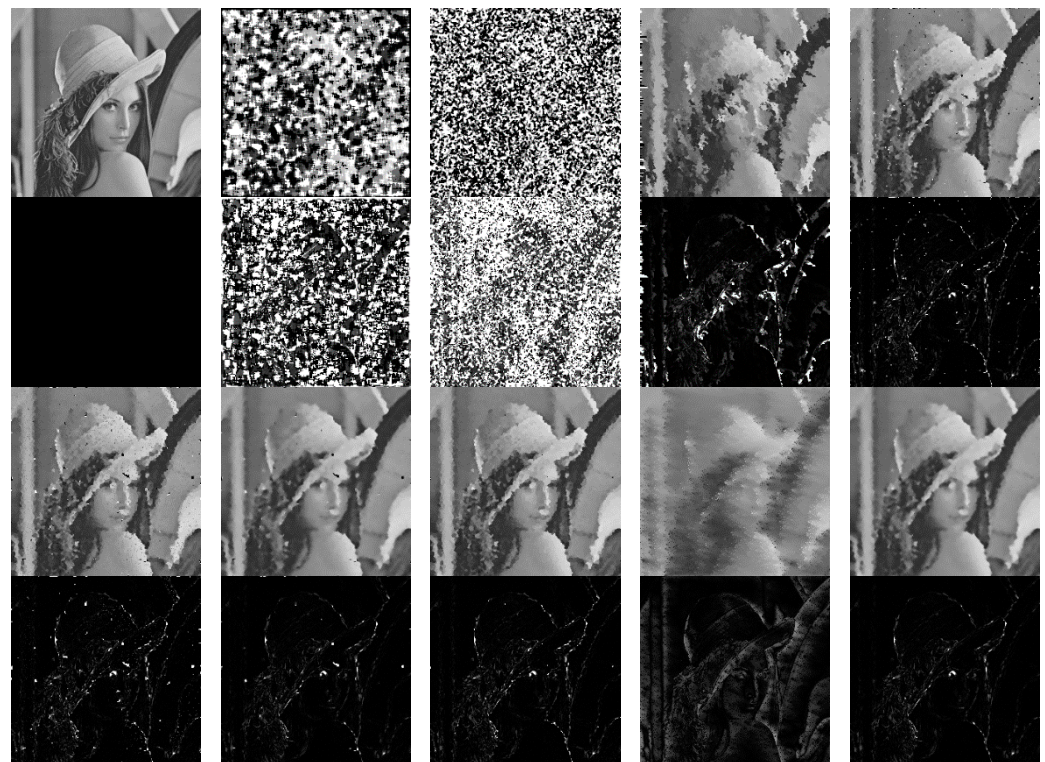


Figure 10. Results of different algorithms for Lena image with 90% noise intensity. The first and third lines are original image, MF, ACWME, DBA, NAFSM, NASEPF, INLM, DAMF, FSAP, and our proposed method. The second and fourth lines are noise image and corresponding error images.

To explore the visual quality of different methods, we showed the reconstructed images with different noise densities. Figures 7 and 8 show the restoration results of various methods for the test images, which were corrupted by salt and pepper noise with 60% and 70% noise density, respectively. The MF and ACWME were also found to fail in restoring the corrupted image. In the reconstructions obtained by DBA and NAFSM, many artifacts exist, which make the contours of the estimated images indistinguishable. The DBA, NAFSM, INLM, DAMF, and FSAP can restore the image with better quality. However, the error noise still exists in the reconstructions. Therefore, the results indicated that the proposed method can preserve details better than other methods.

Figures 9 and 10 show the restoration results of different methods for images that were corrupted by heavy salt and pepper noise with 80% and 90% noise density, respectively. In these figures, the first and third rows were the original images and reconstructed images, respectively. The second and fourth rows were the noised image and corresponding error images, respectively. The SM and DWM filters failed to restore images corrupted by the heavy noise. The visual perception of the reconstructions obtained by MDWM filter was bad because some obvious white and black pixels still exist in the restored images.

Figures 7–10 show that the proposed method achieved better visual quality. Our proposed method can preserve texture regions or edge regions by adaptively calculating the nonlocal region features and estimate the original pixel value. Thereby, the proposed method reduces reconstruction error as far as possible. Moreover, in the existing methods, undesirable artificial artifacts were inevitably produced in the reconstructed images. As we expected, satisfactory visual effects obtained by our developed algorithm were more natural and have fewer artifacts, especially in high noise intensities.

6.5. Effect of Searching Window

In this subsection, we will explore the denoising effect of the proposed method under several searching windows with different size, including 7×7 , 11×11 , 15×15 , and

21×21 . We take four images as test images, in which two 256×256 (i.e., Dog and Zebra images) and two 512×512 images (i.e., boat and Man images), and nine noise levels from $\sigma = 10\text{--}90\%$ are used to generate synthetic noising images. Table 9 presents the average PSNR and SSIM values of the test images under the contribution of different searching windows. In addition to these two quality metrics, another important aspect is the testing speed. Figure 11 shows the average run times of the proposed method with different searching windows for denoising images with 10 noise levels.

From Figure 11 and Table 9, we can see that the methods based on searching windows with 7×7 and 11×11 have high speed. With the noise intensity increases, the speed is lower. Compare with the methods based on searching windows with 7×7 , 11×11 , 15×15 , respectively, the method based on 21×21 window does not achieve obvious higher PSNR and SSIM values. Although more pixels contribute to the estimation, not all pixels are beneficial to restore the corrupted pixels. Moreover, its speed is fairly slow due to the use of more pixels than other windows. Take the 256×256 images for example, the average running time, PSNR and SSIM values achieved corresponding to $\sigma = 10\%$ and searching windows with 7×7 , 11×11 , 15×15 , 21×21 respectively are shown in the following: method based on 7×7 window (10 s, 30.26 db, 0.9681), method based on 11×11 window (26 s, 29.27 db, 0.9682), method based on 15×15 window (48 s, 30.17 db, 0.9686), method based on 21×12 window (92 s, 30.36 db, 0.9684). In the case of $\sigma = 90\%$, the average running time, PSNR and SSIM values achieved by different methods are shown in the following: method based on 7×7 window (77 s, 18.64 db, 0.5687), method based on 11×11 window (187 s, 18.68 db, 0.5557), method based on 15×15 window (346 s, 18.67 db, 0.5489), method based on 21×12 window (684 s, 18.66 db, 0.5440). Therefore, the searching window with 11×11 gains the best performance among all the cases when all of these factors are considered together.

Table 9. Comparison of average PSNR/SSIM values for the test images with four search window sizes (SWS) and nine noise intensities.

| Image Size | SWS | Standard | $\sigma = 10\%$ | $\sigma = 20\%$ | $\sigma = 30\%$ | $\sigma = 40\%$ | $\sigma = 50\%$ | $\sigma = 60\%$ | $\sigma = 70\%$ | $\sigma = 80\%$ | $\sigma = 90\%$ |
|------------------|----------------|----------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 256×256 | 7×7 | PSNR | 30.26 | 27.25 | 26.86 | 25.62 | 24.22 | 23.10 | 21.97 | 20.53 | 18.64 |
| | | SSIM | 0.9681 | 0.9481 | 0.9250 | 0.8939 | 0.8567 | 0.8104 | 0.7539 | 0.6820 | 0.5687 |
| | 11×11 | APSNR | 29.27 | 26.90 | 26.54 | 25.42 | 24.19 | 22.97 | 21.87 | 20.45 | 18.68 |
| | | SSIM | 0.9682 | 0.9472 | 0.9227 | 0.8908 | 0.8510 | 0.8023 | 0.7439 | 0.6687 | 0.5557 |
| | 15×15 | PSNR | 30.17 | 28.27 | 26.57 | 25.5428 | 24.13 | 22.54 | 21.79 | 20.39 | 18.67 |
| | | SSIM | 0.9686 | 0.9484 | 0.9219 | 0.8892 | 0.8482 | 0.7976 | 0.7389 | 0.6625 | 0.5489 |
| | 21×21 | PSNR | 30.36 | 28.19 | 25.81 | 25.40 | 24.05 | 22.78 | 21.73 | 20.35 | 18.66 |
| | | SSIM | 0.9684 | 0.9479 | 0.9207 | 0.8870 | 0.8453 | 0.7945 | 0.7353 | 0.6591 | 0.544 |
| 512×512 | 7×7 | PSNR | 36.74 | 34.52 | 32.47 | 31.00 | 29.66 | 28.40 | 27.12 | 25.83 | 23.82 |
| | | SSIM | 0.9820 | 0.9667 | 0.9462 | 0.9220 | 0.8937 | 0.8608 | 0.8203 | 0.7698 | 0.6020 |
| | 11×11 | PSNR | 35.62 | 34.16 | 32.44 | 30.78 | 29.41 | 28.11 | 26.85 | 25.60 | 23.77 |
| | | SSIM | 0.9818 | 0.9656 | 0.9442 | 0.9178 | 0.8870 | 0.8510 | 0.8077 | 0.7547 | 0.6784 |
| | 15×15 | PSNR | 36.53 | 34.28 | 32.33 | 30.66 | 29.28 | 27.93 | 26.65 | 25.42 | 23.67 |
| | | SSIM | 0.9818 | 0.9654 | 0.9429 | 0.9154 | 0.8835 | 0.8459 | 0.8010 | 0.7472 | 0.6707 |
| | 21×21 | PSNR | 36.69 | 34.48 | 32.24 | 30.55 | 29.15 | 27.76 | 26.46 | 25.23 | 23.52 |
| | | SSIM | 0.9818 | 0.9653 | 0.9419 | 0.9134 | 0.8806 | 0.8416 | 0.7958 | 0.7413 | 0.6648 |

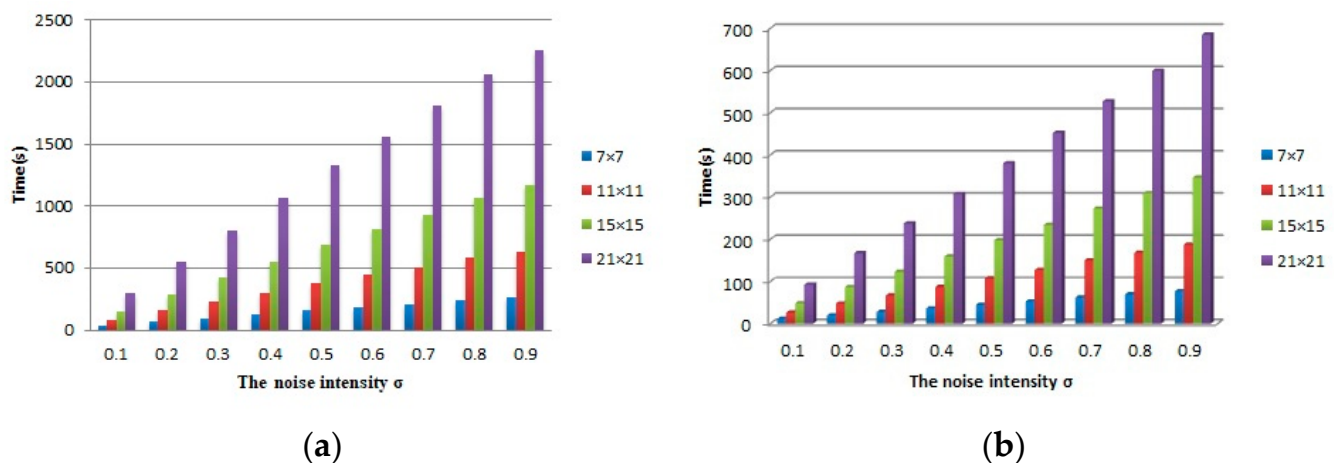


Figure 11. Average running time comparison based on different search window sizes for (a) 512×512 images and (b) 256×256 images. The provided values of search window size are 7×7 , 11×11 , 15×15 , and 21×21 .

7. Conclusions

This paper presented an adaptive nonlocal bilateral filter for salt and pepper noise removal. First, the NAFSM filter was introduced to distinguish the noisy and noise-free pixels and then conduct preliminary filtering on the noisy pixels. Second, an adaptive norm with scale parameters calculated based on local feature was designed to measure the intensity difference between image patches. Finally, the nonlocal thought, the bilateral thought, and the adaptive norm were combined. Then an adaptive nonlocal bilateral filter was designed to suppress the Gaussian–Laplacian mixture noise and further improve the reconstruction quality.

In Section 6, we demonstrate the benefit of our proposed method on the salt and pepper noise removal problem. We first conduct experiments to verify the effectiveness of the adaptive norm adopted in the novel bilateral filter. We have observed that adaptive norm gives comparable or better results compared to the L_1 norm and L_2 norm. Then, we demonstrate that the proposed bilateral filter could effectively weaken the median-type filter errors. Especially for the high noisy intensity ($\sigma = 90\%$), the proposed bilateral filter can obtain satisfactory results. We demonstrate the capability of our proposed method with some state-of-art methods with different noise intensities from $\sigma = 10\%$ to $\sigma = 90\%$. Numerical results illustrated that the proposed method outperformed the state-of-art methods with better visual quality and higher quality metric values. It indicates the effectiveness of our method to use a two-step framework based on the novel-designed adaptive nonlocal bilateral (ANB) filter. Finally, in order to explore the denoising effect of the proposed method under different searching windows, we also compare the denoising results with different window sizes, including 7×7 , 11×11 , 15×15 , and 21×21 . Comprehensively considering the running time, PSNR and SSIM values, the searching window with 11×11 is proven to be a better choice.

Author Contributions: Data curation, H.L.; Formal analysis, S.Z.; Funding acquisition, H.L., N.L., and S.Z.; Methodology, H.L. and S.Z.; Writing—original draft, H.L.; Writing—review & editing, H.L., N.L., and S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: National Natural Science Foundation of China: 61802213; Shandong Provincial Natural Science Found: ZR2020MF039; National Key Research and Development Project: 2019YFB1404701; The Major Science and Technology Innovation Projects of Key R&D Programs of Shandong Province in 2019 (Development and Industrialization of Pathological Artificial Intelligence Application Software): 2019JZZY010108; Qilu University of Technology (Shandong Academy of Sciences) Young doctor Cooperation Fund Project: 2019BSHZ009.

Institutional Review Board Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yuan, G.; Ghanem, B. TV: A Sparse Optimization Method for Impulse Noise Image Restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 352–364. [\[CrossRef\]](#)
2. Jin, L.; Zhu, Z.; Song, E.; Xu, X. An Effective Vector Filter for Impulse Noise Reduction Based on Adaptive Quaternion Color Distance Mechanism. *Signal Process.* **2019**, *155*, 334–345. [\[CrossRef\]](#)
3. Veerakumar, T.; Subudhi, B.N.; Esakkirajan, S.; Pradhan, P.K. Iterative Adaptive Unsymmetric Trimmed Shock Filter for High-density Salt-and-pepper Noise Removal. *Circuits Syst. Signal Process.* **2019**, *38*, 2630–2652. [\[CrossRef\]](#)
4. Fu, B.; Zhao, X.; Song, C.; Li, X.; Wang, X. A Salt and Pepper Noise Image Denoising Method Based on the Generative Classification. *Multimed. Tools Appl.* **2019**, *78*, 12043–12053. [\[CrossRef\]](#)
5. Chen, F.; Huang, M.; Ma, Z.; Li, Y.; Huang, Q. An Iterative Weighted-Mean Filter for Removal of High-Density Salt-and-Pepper Noise. *Symmetry* **2020**, *12*, 1990. [\[CrossRef\]](#)
6. Zeng, X.; Yang, L. A Robust Multiframe Super-resolution Algorithm Based on Half-quadratic Estimation with Modified BTV Regularization. *Digit. Signal Process. Rev. J.* **2013**, *23*, 98–109. [\[CrossRef\]](#)
7. Zhang, X.-M.; Kang, Q.; Cheng, J.-F.; Wang, X. Adaptive Four-dot Median Filter for Removing 1–99% Densities of Salt and-pepper Noise in Images. *J. Inf. Technol. Res.* **2018**, *11*, 47–61. [\[CrossRef\]](#)
8. Rhee, K.H. Improvement Feature Vector: Autoregressive Model of Median Filter Error. *IEEE Access* **2019**, *7*, 77524–77540. [\[CrossRef\]](#)
9. Storath, M.; Weinmann, A. Fast Median Filtering for Phase or Orientation Data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 639–652. [\[CrossRef\]](#)
10. Gao, H.; Hu, M.; Gao, T.; Cheng, R. Robust Detection of Median Filtering Based on Combined Features of Difference Image. *Signal Process. Image Commun.* **2019**, *72*, 126–133. [\[CrossRef\]](#)
11. Zhang, Z.; Han, D.; Dezert, J.; Yang, Y. A New Adaptive Switching Median Filter for Impulse Noise Reduction with Pre-detection Based on Evidential Reasoning. *Signal Process.* **2018**, *147*, 173–189. [\[CrossRef\]](#)
12. Zhu, Z.; Jin, L.; Song, E.; Hung, C.-C. Quaternion Switching Vector Median Filter Based on Local Reachability Density. *IEEE Signal Process. Lett.* **2018**, *25*, 843–847. [\[CrossRef\]](#)
13. Chanu, P.R.; Singh, K.M. A Two-stage Switching Vector Median Filter Based on Quaternion for Removing Impulse Noise in Color Images. *Multimed. Tools Appl.* **2019**, *78*, 15375–15401. [\[CrossRef\]](#)
14. Sanaee, P.; Moallem, P.; Razzazi, F. An Interpolation Filter Based on Natural Neighbor Galerkin Method for Salt and Pepper Noise Restoration with Adaptive Size Local Filtering Window. *Signal Image Video Process.* **2019**, *13*, 895–903. [\[CrossRef\]](#)
15. Varatharajan, R.; Vasanth, K.; Gunasekaran, M.; Priyan, M.; Gao, X.Z. An Adaptive Decision Based Kriging Interpolation Algorithm for the Removal of High Density Salt and Pepper Noise in Images. *Comput. Electr. Eng.* **2018**, *70*, 447–461. [\[CrossRef\]](#)
16. Toh, K.K.V.; Isa, N.A.M. Noise Adaptive Fuzzy Switching Median Filter for Salt-and-pepper Noise Reduction. *IEEE Signal Process. Lett.* **2010**, *17*, 281–284. [\[CrossRef\]](#)
17. Ibrahim, H.; Kong, N.S.P.; Foo, T.F. Simple Adaptive Median Filter for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Trans. Consum. Electron.* **2008**, *54*, 1920–1927. [\[CrossRef\]](#)
18. Toh, K.K.V.; Ibrahim, H.; Mahyuddin, M.N. Salt-and-pepper Noise Detection and Reduction Using Fuzzy Switching Median Filter. *IEEE Trans. Consum. Electron.* **2008**, *54*, 1956–1961. [\[CrossRef\]](#)
19. Wang, Y.; Wang, J.; Song, X.; Han, L. An Efficient Adaptive Fuzzy Switching Weighted Mean Filter for Salt-and-pepper Noise Removal. *IEEE Signal Process. Lett.* **2016**, *23*, 1582–1586. [\[CrossRef\]](#)
20. Kumar, S.V.; Nagaraju, C. T2FCS Filter: Type 2 Fuzzy and Cuckoo Search-based Filter Design for Image Restoration. *J. Vis. Commun. Image Represent.* **2019**, *58*, 619–641. [\[CrossRef\]](#)
21. González-Hidalgo, M.; Massanet, S.; Mir, A.; Ruiz-Aguilera, D. Improving Salt and Pepper Noise Removal Using a Fuzzy Mathematical Morphology-based Filter. *Appl. Soft Comput. J.* **2018**, *63*, 167–180. [\[CrossRef\]](#)
22. Khosravy, M.; Gupta, N.; Marina, N.; Sethi, I.K.; Asharif, M.R. Morphological Filters: An Inspiration from Natural Geometrical Erosion and Dilation. In *Nature Inspired Computer Optimization*; Springer: Cham, Switzerland, 2017; pp. 349–379.
23. Sedaaghi, M.H.; Daj, R.; Khosravi, M. Mediated Morphological Filters. In *Proceedings of the 2001 International Conference on Image Processing, Thessaloniki, Greece, 7–10 October 2001*; 2001; pp. 692–695.
24. Khosravi, M.; Sedaaghi, M.H. Impulsive Noise Suppression of Electrocardiogram Signals with Mediated Morphological Filters. In *Proceedings of the 11th Iranian Conference on Biomedical Engineering, Teheran, Iran, 18 February 2004*.
25. Khosravy, M.; Asharif, M.R.; Sedaaghi, M.H. Morphological Adult and Fetal ECG Preprocessing: Employing Mediated Morphology. *IEICE Med. Imaging* **2007**, *107*, 363–369.
26. Khosravy, M.; Asharif, M.R.; Sedaaghi, M.H. Medical Image Noise Suppression: Using Mediated Morphology. *IEICE Tech. Rep.* **2008**, *107*, 265–270.
27. Mafi, M.; Rajaei, H.; Cabrerizo, M.; Malek, A. A Robust Edge Detection Approach in the Presence of High Impulse Noise Intensity through Switching Adaptive Median and Fixed Weighted Mean Filtering. *IEEE Trans. Image Process.* **2018**, *27*, 5475–5490. [\[CrossRef\]](#)

28. Balasubramanian, S.; Kalishwaran, S.; Muthuraj, R.; Ebenezer, D.; Jayaraj, V. An Efficient Non-linear Cascade Filtering Algorithm for Removal of High Density Salt and Pepper Noise in Image and Video Sequence. In Proceedings of the IEEE International Conference Control, Automation, Communication and Energy Conservation (INCACEC), Perundurai, Erode, India, 4–6 June 2009; pp. 1–6.
29. Raza, M.T.; Sawant, S. High Density Salt and Pepper Noise Removal through Decision Based Partial Trimmed Global Mean Filter. In Proceedings of the IEEE International Conference Engineering (NUICONE), Ahmedabad, Gujarat, India, 6–8 December 2012; pp. 1–5.
30. Esakkirajan, S.; Veerakumar, T.; Subramanyam, A.N.; Chand, C.H.P. Removal of High Density Salt and Pepper Noise through Modified Decision Based Unsymmetrical Trimmed Median Filter. *IEEE Signal Process. Lett.* **2011**, *18*, 287–290. [[CrossRef](#)]
31. Wang, X.; Shen, S.; Shi, G.; Xu, Y.; Zhang, P. Iterative Non-local Means Filter for Salt and Pepper Noise Removal. *J. Vis. Commun. Image Represent.* **2016**, *38*, 440–450. [[CrossRef](#)]
32. Veerakumar, T.; Subudhi, B.N.; Esakkirajan, S. Empirical Mode Decomposition and Adaptive Bilateral Filter Approach for Impulse Noise Removal. *Expert Syst. Appl.* **2019**, *121*, 18–27. [[CrossRef](#)]
33. Mafi, M.; Martin, H.; Cabrerizo, M.; Andrian, J.; Barreto, A.; Adjouadi, M. A Comprehensive Survey on Impulse and Gaussian Denoising Filters for Digital Images. *Signal Process.* **2019**, *157*, 236–260. [[CrossRef](#)]
34. Nair, P.; Chaudhury, K.N. Fast High-dimensional Bilateral and Nonlocal Means Filtering. *IEEE Trans. Image Process.* **2019**, *28*, 1470–1481. [[CrossRef](#)] [[PubMed](#)]
35. Karthikeyan, P.; Vasuki, S. Efficient Decision Based Algorithm for the Removal of High Density Salt and Pepper Noise in Images. *J. Commun. Technol. Electron.* **2016**, *61*, 963–970. [[CrossRef](#)]
36. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
37. Chen, T.; Wu, H.R. Adaptive Impulse Detection Using Center Weighted Median Filters. *IEEE Signal Process. Lett.* **2001**, *8*, 1–3. [[CrossRef](#)]
38. Jiang, D.-S.; Li, X.-B.; Wang, Z.-L.; Liu, C. An Efficient Edge-preserving Approach Based on Adaptive Fuzzy Switching Median Filter. In Proceedings of the 2011 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering, Xi'an, China, 17–19 June 2011; pp. 952–955.
39. Erkan, U.; Gökrem, L.; Enginoğlu, S. Different Applied Median Filter in Salt and Pepper Noise. *Comput. Electr. Eng.* **2018**, *70*, 789–798. [[CrossRef](#)]
40. Srinivasan, K.S.; Ebenezer, D. A New Fast and Efficient Decision-based Algorithm for Removal of High-density Impulse Noises. *IEEE Signal Process. Lett.* **2007**, *14*, 189–192. [[CrossRef](#)]
41. Singh, V.; Dev, R.; Dhar, N.K.; Agrawal, P.; Verma, N.K. Adaptive Type-2 Fuzzy Approach for Filtering Salt and Pepper Noise in Grayscale Images. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 3170–3176. [[CrossRef](#)]