Research Article

NAMF: A Nonlocal Adaptive Mean Filter for Removal of Salt-and-Pepper Noise

Houwang Zhang ^{1,2}, Yuan Zhu ^{1,2} and Hanying Zheng ^{1,2}

¹School of Automation, China University of Geosciences, Wuhan, China

²Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan, China

Correspondence should be addressed to Yuan Zhu; zhuyuan@cug.edu.cn

Received 27 July 2020; Accepted 22 February 2021; Published 10 March 2021

Academic Editor: Xiaojie Guo

Copyright © 2021 Houwang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a novel algorithm called a Nonlocal Adaptive Mean Filter (NAMF) for removing salt-and-pepper (SAP) noise from corrupted images is presented. We employ an efficient window detector with adaptive size to detect the noise. The noisy pixel is then replaced by the combination of its neighboring pixels, and finally, a SAP noise based nonlocal mean filter is used to reconstruct the intensity values of noisy pixels. Extensive experimental results demonstrate that NAMF can obtain better performance in terms of quality for restoring images at all levels of SAP noise.

1. Introduction

Digital images are often corrupted by noises in the process of image acquisition and transmission [1, 2]. Image noise is a random change in the brightness or color information in an image, which will bring troubles for image processing and computer vision, even causing some seriously ill-posed problems [3]. Hence, image denoising has been taken as one of the most important tasks in image processing [4, 5]. As a preprocessing step in image processing, image denoising can protect edges, textures, and other details in the image [6]. Generally, image noises can be divided into Gaussian noise, Poisson noise, and Salt-and-Pepper (SAP) noise in terms of discreteness and randomness. SAP noise commonly exists in natural images, and the pixels contaminated by SAP noise take the maximum or minimum value and can be represented as black or white points [7, 8]. Considering the universality of SAP noise, we focus our attention on the SAP noise and develop a denoising method that provides some help for further image analysis.

To remove SAP noise, lots of computational methods have been proposed. Among them, median filter (MF) and adaptive median filter (AMF) [9] are the two most popular methods in the early stage. MF can restore image details well

under low noise intensity, but it performs poorly when noise intensity is high [10]. AMF adopts the measure of the window with adaptive size, which makes it perform well in high noise intensity [11]. In addition, based on AMF, noise adaptive fuzzy switching median filter (NAFSMF) recognizes SAP noise by analyzing the histogram of noisy images and takes a fuzzy method to denoise [12]. Adaptive weighted mean filter (AWMF) uses two successive windows to detect noisy pixels and processes them with a weighted mean filter [13]. The method proposed in [14] is based on NAFSMF and AWMF, and it uses a new adaptive fuzzy switching weighted mean filter to remove SAP noise. In [15], researchers proposed a method, based on pixel density filter (BPDF), to remove SAP noise through searching the repeated numbers of the pixels, and achieves a good performance under low SAP noise intensity. Different applied median filter (DAMF) is proposed for removing SAP noise at all densities [16]. In [17], adaptive frequency median filter (AFMF) is proposed to remove the SAP noise, which employs frequency median to restore gray values of the corrupted pixels. The adaptive switching weight mean filter (ASWMF) proposed in [18] repairs noisy pixels by assigning a value of a switching weight mean to the gray value of the center pixel of an adaptive window.

The processes of the above denoising methods can be divided into two stages, the detection of noisy pixels and the restoration of noisy pixels. For the latter one, the restoration of SAP noise is using the rest information (uncontaminated pixels) to repair the absent information (contaminated pixels). As Figure 1(b) shows, the limitation of existing state-of-the-art methods is that under high SAP noise level, the boundary of the restored image is jagged and the details are blurred. When SAP noise intensity is too high, the image loses too much information. The consequence is that less information is available to be used for restoring the image.

To solve the above problem, in this paper, we propose a nonlocal adaptive mean filter (NAMF) to remove SAP noise. NAMF can be divided into two stages: (1) SAP noise detection; (2) SAP noise elimination. Firstly, in the stage of SAP noise detection, we find out pixels whose gray values are equal to the global maximum or minimum gray value of the image. We then take them as noisy candidates and use a window with adaptive size to confirm them. For the possible noisy pixel, we calculate the proportion of the pixels with the same value of candidate in the window and then filter it by a threshold. If it is smaller than the threshold, the candidate is regarded as a noisy pixel. Otherwise, it is noiseless and will not be processed. Secondly, in the stage of SAP noise elimination, the noisy pixel will be replaced by the mean of its neighboring pixels. Finally, we use a SAP noise based nonlocal mean filter to further restore it. The main contributions of this paper can be concluded as follows:

- (i) A proportion based method is proposed to distinguish the noise pixels and texture pixels in the SAP

noise detection, which can effectively avoid noiseless texture pixels being processed as noisy pixels.

- (ii) An improved nonlocal mean method based on the characteristics of SAP noise is developed to further restore noisy images, which can further enhance the quality of restored images.

In the experiments, we use 16 typical images and 40 test images collected from the TEST IMAGES Database [19] to evaluate our method. The experimental results demonstrate that NAMF outperforms the existing state-of-the-art methods at both high SAP noise levels and low SAP noise levels.

2. Related Works

In this section, we will firstly review traditional MF based methods used for SAP noise removal and then we introduce the classical nonlocal means method.

2.1. MF Based Methods for SAP Noise Elimination. MF is the most commonly used algorithm to remove pulse interference and SAP noise. The main idea is that all pixels are replaced by neighborhood median pixels whether pixels are noiseless or not. Detailedly, let $\mathbf{X}$ be an original 8-bit gray-level image with a size of $M \times N$, and $x_{i,j}$ be the gray value of the pixel at the location (i, j) . Then, for MF with a filter size of 3×3 , the new gray value $\tilde{x}_{i,j}$ of the pixel at the location (i, j) can be calculated by the following equation:

$$\tilde{x}_{i,j} = \text{median}(x_{i-1,j-1}, x_{i-1,j}, x_{i-1,j+1}, x_{i,j-1}, x_{i,j}, x_{i,j+1}, x_{i+1,j-1}, x_{i+1,j}, x_{i+1,j+1}). \quad (1)$$

Motivated by this idea, many MF methods have been proposed. However, the large distance between the noisy pixel and its neighborhood may result in a large difference between the median value pixel and the original noiseless pixel, which would cause the denoised image to be degraded [14]. In [13], the median value of neighborhoods is equipped with different weights, which are calculated by the distance from the center noisy pixel. Besides, when noise density is too high, for some noisy pixels, there are no noiseless neighborhoods for restoring. In [14], to tackle the problem, the researchers provide a method by using neighborhoods in the restored image to calculate the median value.

2.2. Nonlocal Mean (NLM) Method. The NLM method combined the idea of the Yaroslavsky filter [20] and bilateral filter [21]. For the noisy pixel, NLM uses the intensity patch feature instead of single pixel feature to compute the denoised mean value [22].

Given a pixel $x \in \mathbf{I}$, let $N(x)$ denote an image block centered at x . Then, NLM calculates a weighted average of all the pixels in the searching window B by the following equations:

$$\tilde{x} = \frac{\sum_{y \in B} \mathcal{Y} * u(x, y)}{\sum_{y \in B} u(x, y)}, \quad (2)$$

$$u(x, y) = e^{-d(x, y)/h^2}, \quad (3)$$

$$d(x, y) = \|N(x) - N(y)\|_{2,a}^2, \quad (4)$$

where $\tilde{x}$ is the restored gray value of x , while $N(x)$ denotes a block centered at x , and h is the smoothing parameter. $u(x, y)$ denotes the similarity between pixels x and y which depends on the weighted Euclidean distance between two image blocks $N(x)$ and $N(y)$, $\|\cdot\|_2$ is the Euclidean norm, and a is the standard deviation of the Gaussian kernel with zero mean and standard deviation, for details about these, please refer to [22].

3. The Proposed Nonlocal Adaptive Mean (NAMF) Filter

In this section, we will present the details of our proposed nonlocal adaptive mean (NAMF) filter. Same as the settings of many traditional SAP noise removal filters, NAMF can be

FIGURE 1: (a) “Lena” corrupted by SAP noise ratio of 10%, and the gray values of noisy candidates in E and F are both 0. (b) The result of the method proposed [14] for “Lena” corrupted by SAP noise ratio of 90%.

divided into two stages: (1) SAP noise detection; (2) SAP noise elimination. Details are presented as follows.

3.1. SAP Noise Detection. In accordance with the mathematical notation as the previous section, $X = (x_{i,j}) \in \mathbb{R}^{M \times N}$ and $Y = (y_{i,j}) \in \mathbb{R}^{M \times N}$ represent the original 8-bit gray-level image and noisy image corrupted by SAP noise, respectively. $x_{i,j}$ and $y_{i,j}$ represent the gray value of the pixel at the location (i, j) of X and Y , respectively, where $(i, j) \in \Lambda \equiv \{1, \dots, M\} \times \{1, \dots, N\}$.

In a corrupted image, the value of a “salt” pixel equals the maximum gray value 255, and the value of a “pepper” pixel equals the minimum value 0. Thus, $y_{i,j}$ is defined by the following equation:

$$y_{i,j} = \begin{cases} 255 \text{ or } 0, & \text{with probability } \alpha, \\ x_{i,j}, & \text{with probability } 1 - \alpha, \end{cases} \quad (5)$$

where α is the density of SAP noise of Y , α is unknown in practical problems, here we just use α to represent the degree of noise. And in the process of image denoising, we use $S_{i,j}(w)$ to represent a $(2w+1) \times (2w+1)$ window centered at (i, j) with the radius w .

Considering the characteristics of SAP noise, pixel $y_{i,j}$ corrupted by SAP noise is 0 or 255. That is to say, noisy pixel candidate $y_{i,j}$ only has 2 possible values: $y_{\min} = 0$ and $y_{\max} = 255$. Following is a prior decision condition in noise detection.

$$o_{i,j} = \begin{cases} 1, & y_{i,j} = 0 \text{ or } 255, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where $\mathbf{O} = (o_{i,j}) \in \mathbb{R}^{M \times N}$ is an indicator matrix with binary value. $o(i, j) = 1$ means that pixel $y_{i,j}$ is the noisy pixel candidate, while $o(i, j) = 0$ that means pixel $y_{i,j}$ is noiseless. For a natural image, the pixels with high or low value are also possible to be the texture of the image. For example, as shown in Figure 1(a), E and F are both candidates with a gray value of 0 (black). However, F has a larger possibility to be a noisy pixel, and E has a larger possibility to be a texture pixel. Therefore, it is very necessary to further confirm these pixels with maximum or minimum value.

In fact, in natural image, pixels are supposed to change smoothly, and adjacent pixels tend to have similar values.

The value of pixels in an image with SAP noise may change suddenly. Motivated by these facts, we propose an adaptive SAP noise detection method by considering the neighborhood to identify noise candidates as noisy pixels or not. Its procedures can be divided into two steps:

- (i) *Step 1.* For the pixel $y_{i,j}$ with $o_{i,j} = 1$, we calculate $S_{i,j}^{\text{sum}}(w)$ within an adaptive searching window $S_{i,j}(w)$. The radius w of window $S_{i,j}(w)$ is initially set to 1. If the condition is met ($w = w_{\max}$ or $S_{i,j}^{\text{sum}}(w) > 0$), the w is just what we need. If not, then $w + 1$ and continue to compute, where $S_{i,j}^{\text{sum}}(w)$ is the number of pixels within $S_{i,j}(w)$ which are not equal to $y_{\min}$ and $y_{\max}$, $w_{\max}$ is the maximum size of window. If $S_{i,j}^{\text{sum}}(w) > 0$, we take the pixel $y_{i,j}$ as a noisy pixel, the detection for $y_{i,j}$ ends. Otherwise, we go to Step 2.
- (ii) *Step 2.* If $S_{i,j}^{\text{sum}}(w) = 0$, $y_{i,j}$ maybe considered as a texture pixel. Then, we calculate the proportion $\rho = S_{i,j}^{\text{num}}(w) / ((2w+1) \times (2w+1))$, where $S_{i,j}^{\text{num}}(w)$ is the number of pixels owning the same value as the candidate pixel $y_{i,j}$ in the window $S_{i,j}(w)$. After that, we set a threshold T to identify the noisy pixel. If $\rho < T$, the candidate pixel $y_{i,j}$ is regarded as a noisy pixel, else it is noiseless.

For the pixel $y_{i,j}$, if it is finally detected as noisy, we mark it with the discriminant matrix $\mathbf{L} = l(i, j) \in \mathbb{R}^{M \times N}$, and $l(i, j) = 1$, else $l(i, j) = 0$ and will not be processed.

3.2. SAP Noise Elimination. We restore noisy pixels in two steps. Let $\mathbf{Z}$ represent the initially restored image, $\hat{\mathbf{Z}}$ represent the final output image. Before processing, we initialize $\mathbf{Z}$ with $\mathbf{Z} = \mathbf{Y}$.

Firstly, when a pixel $y_{i,j}$ is detected as a noisy pixel, we use $S_{i,j}^{\text{mean}}(w)$ to restore it. The calculation of $S_{i,j}^{\text{mean}}(w)$ is based on $S_{i,j}^{\text{sum}}(w)$ as shown in equation (7). When $S_{i,j}^{\text{sum}}(w) \neq 0$, $S_{i,j}^{\text{mean}}(w)$ is the mean of the noiseless pixels in $S_{i,j}(w)$, otherwise it is the mean of three processed neighboring pixels of $y_{i,j}$ in $\mathbf{Z}$. Unlike the four neighbors adopted in [14], the utilization of four neighboring pixels will lead to residual noisy pixels on the boundary, as shown in Figure 1(b) (even if the image boundary is expanded during the process, some noisy pixels located in boundary can still not be restored). Hence, we

select three processed neighboring pixels in $\mathbf{Z}$, i.e., $S_{i,j}^{\text{mean}}(w) = (z_{i-1,j-1} + z_{i-1,j} + z_{i,j-1})/3$, and its illustration is shown in Figure 2.

$$S_{i,j}^{\text{mean}}(w) = \begin{cases} \frac{\sum_{(e,f) \in S_{i,j}(w)} (1 - l(e, f)) * y_{e,f}}{\sum_{(e,f) \in S_{i,j}(w)} (1 - l(e, f))}, & S_{i,j}^{\text{sum}}(w) \neq 0, \\ z_{i-1,j-1} + z_{i-1,j} + z_{i,j-1}/3, & \text{otherwise,} \end{cases} \quad (7)$$

Secondly, considering the problem that restoration will be hard under high noise intensity, we introduce the non-local mean (NLM) method [22, 23] to further restore noisy pixels. NLM can restore noisy pixels by using all neighbors

instead of part of neighbors, which are detected as noiseless. In this way, we can use NLM to further enhance the restoration effect. Considering that the original NLM is designed for Gaussian noise or Speckle noise, which are quite different from SAP noise, we modify it based on the characteristics of SAP noise and make it applicable for our method.

In the noise detection stage, a pixel is identified and marked by the discriminant matrix $\mathbf{L}$. $l(i, j) = 1$ means the pixel $y_{i,j}$ is identified as a noisy pixel, otherwise noiseless. After the restoration of the noisy pixel $y_{i,j}$, we get $z_{i,j}$. The modified NLM method is used to further restore $z_{i,j}$. We use $\hat{z}_{i,j}$ to represent the value of the further restored $z_{i,j}$, and it can be calculated as follows:

$$\hat{z}_{i,j} = \begin{cases} \frac{1}{C} \sum_{z_{e,f} \in B(z_{i,j}, r)} z_{e,f} * u(z_{i,j}, z_{e,f}), & l(i, j) = 1, \\ z_{i,j}, & l(i, j) = 0, \end{cases} \quad (8)$$

$$C = \sum_{z_{e,f} \in B(z_{i,j}, r)} u(z_{i,j}, z_{e,f}), \dots, u(z_{i,j}, z_{e,f}) = \begin{cases} e^{-d(z_{i,j}, z_{e,f})/h^2}, & e \neq i \text{ or } f \neq j, \\ 0, & e = i \text{ and } f = j, \end{cases} \quad (9)$$

$$d(z_{i,j}, z_{e,f}) = \|N(z_{i,j}) - N(z_{e,f})\|_{2,a}^2, \dots, h = \left(\frac{\bar{L}}{M \times N} \right)^2 * \beta_2 + \frac{\bar{L}}{M \times N} * \beta_1 + \beta_0, \quad (10)$$

where $B(z_{i,j}, r)$ represents a searching window of size $(2r + 1) \times (2r + 1)$ centered at $z_{i,j}$, and $u(z_{i,j}, z_{e,f})$ represents the weight of pixel $z_{e,f}$ in $B(z_{i,j}, r)$. $N(z_{i,j})$, also called similarity window, is a square block centered at $z_{i,j}$, so is $N(z_{e,f})$. As shown in equation (10), the similarity between $z_{i,j}$ and $z_{e,f}$ is measured by the Gaussian weighted Euclidean distance $d(z_{i,j}, z_{e,f})$ between $N(z_{i,j})$ and $N(z_{e,f})$, where a is the standard deviation of the Gaussian kernel, and h is the smoothing parameter for NLM.

When processing the noisy pixel $z_{i,j}$, the original NLM assigns the weight based on the similarity. That is to say, the weight of pixel $z_{i,j}$ itself is the largest. Different from the original NLM, in our method, the noisy pixel to be processed will not participate in the process of NLM. Thus the weight of the pixel $z_{i,j}$ should be set as 0, as shown in equation (9).

In the NLM algorithm, the higher the noise intensity is, the larger the smoothing parameter h should be. But the intensity of noise is not easy to be confirmed. Considering that SAP noise can be significantly detected, we use the intensity of SAP noise to confirm h . As shown in equation (10), $\bar{L}$ represents the total number of non-zero elements in the discriminant matrix $\mathbf{L}$, that is, the more noisy pixels are detected, the larger h should be; β_0 , β_1 and β_2 are the parameters used to fit h .

Due to the high computational cost of the NLM algorithm, here we introduce a kind of fast implementation of the NLM

algorithm [24] based on the computation of patch distances using sums of lines to accelerate our NAMF algorithm. The details of the proposed NAMF are shown in Algorithm 1.

4. Experimental Results

In the experiments, NAMF are compared with 8 state-of-the-art methods: AMF [9], NAFSMF [12], AWMF [13], the method proposed in [14], BPDF [15], DAMF [16], AFWM [17], and ASWMF [18]. Sixteen typical images (Barbara, Elaine, Goldhill, Lena, Man, Peppers, Yacht, and Zelda, Baboon, House, Boat, Cameraman, Einstein, Face, Straw, and Couple) and 40 test images in the TEST IMAGES Database [19] are chosen for the experiments.

We use 4 typical image quality metrics, peak signal-to-noise ratio (PSNR) [25], structural similarity (SSIM) [26], mean absolute error (MAE) [27], and image enhancement factor (IEF) [28] to evaluate the experimental results. For an image U and an image V with the same size of $M \times N$, PSNR can be calculated by equation (11).

$$\text{PSNR}(U, V) = 10 * \log_{10}(255^2 / \text{MSE}), \quad (11)$$

$$\text{MSE} = \frac{1}{M \times N} \sum_i \sum_j (u_{i,j} - v_{i,j})^2,$$


```

/* STAGE 1 */
Compute the indicator matrix O
For each pixel  $(i, j) \in \Lambda$  in the noisy image Y and the initially restored image Z, do
(1) If  $o_{i,j} == 0, l(i, j) = 0, z_{i,j} = y_{i,j}$ , break;
    Otherwise, go to step 2).
(2) Initialize  $w = 1, h = 1, w_{\max} = 7$ .
(3) Compute  $S_{i,j}^{\text{sum}}(w)$  until  $w == w_{\max}$  or  $S_{i,j}^{\text{sum}}(w) > 0$ ;
    Otherwise,  $w = w + h$  and repeat step 3).
(4) If  $S_{i,j}^{\text{sum}}(w) > 0, l(i, j) = 1, z_{i,j} = S_{i,j}^{\text{mean}}(w)$ , break;
    Otherwise, go to step 5).
(5) Compute  $\rho$  If  $\rho < T, l(i, j) = 1, z_{i,j} = S_{i,j}^{\text{mean}}(w)$ ;
    Otherwise  $l(i, j) = 0, z_{i,j} = y_{i,j}$ .
/* STAGE 2 */
Compute  $h$ .
For each pixel  $z_{i,j}$  in image Z and  $\tilde{z}_{i,j}$  in output image  $\tilde{Z}$ , do
(6) If  $l(i, j) == 1, \tilde{z}_{i,j} = 1/c \sum_{z_e, f \in B(z_{i,j}, r)} z_e, f * u(z_{i,j}, z_{e,f})$ ;
    Otherwise  $\tilde{z}_{i,j} = z_{i,j}$ .

```

ALGORITHM 1: NAMF.

FIGURE 2: **Y** is a noisy image, **Z** is an image being restored, the red pixel in **Z** is prepared to process, the yellow pixels are pixels that have been restored, other blue pixels in **Z** are pixels that remained to be restored. When processing noisy pixels $y_{i,j}$, neighborhoods in the window are all noisy, i.e., $S_{i,j}^{\text{sum}}(w) = 0$. So here we select three neighboring pixels $z_{i-1,j-1}$, $z_{i-1,j}$ and $z_{i,j-1}$ in the restored image **Z** for restoring pixel $y_{i,j}$ in the original image **Y**.

where MSE is the mean square error of two images, and $u_{i,j}$, $v_{i,j}$ are pixels of image **U** and **V**, respectively. SSIM for image **U** and **V** can be defined as follows:

$$\text{SSIM}(U, V) = \frac{(2\mu_u\mu_v + c1) * (2\sigma_{u,v} + c2)}{(\mu_u^2 + \mu_v^2 + c1) * (\sigma_u^2 + \sigma_v^2 + c2)}, \quad (12)$$

where μ_u and μ_v are the average intensities of image **U** and **V**, respectively. σ_u and σ_v are standard deviations, $\sigma_{u,v}$ is the covariance, $c1$ and $c2$ are some constants. Here $c1$ and $c2$ are set to be $(0.01 * 255)^2$ and $(0.03 * 255)^2$ as in [26], respectively. MAE can be computed as shown in the following equation:

$$\text{MAE}(U, V) = \frac{1}{M \times N} \sum_i \sum_j |u_{i,j} - v_{i,j}|, \quad (13)$$

Different from traditional image quality metrics, IEF is designed for evaluating the respiring effects of the denoising method. IEF can be obtained by the following equation:

$$\text{IEF} = \frac{\sum_i \sum_j (\ddot{I}_{i,j} - \dot{I}_{i,j})^2}{\sum_i \sum_j (\dot{I}_{i,j} - I_{i,j}^o)^2}, \quad (14)$$

where $\dot{I}$, $\ddot{I}$, I^o denote the original image, the noisy image, and the repaired image, respectively. In this paper, we set $\beta_2 = 2.2186$, $\beta_2 = 2.2186$, and $\beta_0 = 4.5595$ to fit h . The size of the searching window is 5×5 . And the size of the similarity window is 41×41 . Through the test, we finally take the threshold $T = 0.8$ for our method. Other methods keep the default parameters. The experiments are performed on a personal computer with Intel Core i7 2.2 GHz processor and 16 GB RAM.

Figure 3 shows the restored results of all methods for “Barbara” with SAP noise ratio of 10%. By observing the enlarged area in Figure 3, we can see that AMF, AWMF, BPDF, and ASWMF cannot restore the details of the original image very well. And the result of our proposed NAMF is nearly the same as the original image.

Figure 4 shows the restored results for “Lena” with SAP noise ratio of 90%. In the enlarged area in Figure 4, it can be found that the performances of AMF, NAFSMF, BPDF,

FIGURE 3: Experimental results of different methods for “Barbara” with SAP noise ratio of 10%. (a) Noisy image. (b) AMF. (c) NAFSMF. (d) AWMF. (e) [14]. (f) BPDF. (g) DAMF. (h) AFMF. (i) ASWMF. (j) NAMF. (k) Original image.

AFMF, and ASWMF are very poor. And the restored images by other methods are also very blurred, while the result of our method looks more natural and smooth.

The curves of average PSNR, SSIM, MAE, and IEF are shown in Figure 5. Figure 5(a) illustrates that NAMF obtains the highest PSNR under both low and high noise intensity, and the PSNR of our method is much higher than the results of other methods. Figure 5(b) shows that the SSIM curves of most methods are basically the same under low SAP noise intensity. However, with the increase of noise level, the superiority of NAMF is getting more obvious. When the noise ratio exceeds 30%, the SSIM obtained by NAMF is significantly higher than other methods. Figure 5(c) gives the MAE curves of all methods. We find that NAMF owns the lowest MAE at all SAP noise levels. The IEF curves of all methods are given in Figure 5(d). It is clear that our method can obtain the highest IEF score at all SAP noise levels. Furthermore, our NAMF can significantly outperform other methods in the low SAP noise level.

Figure 6 illustrates the average running time of different methods at all noise levels. With the increase of noise intensity, the running time of all methods except AWMF and AFMF increases. Clearly, the average running time of the method proposed in [14] is the shortest. As shown in the enlarged area of Figure 6, although the rank of NAMF is in the middle, its processing speed is superior to NAFSMF and similar to DAMF under high noise intensity.

Besides, to further illustrate the function of our proposed SAP based NLM method, we make the ablation experiments to show its effectiveness. As shown in Figure 7, we compare the NAMF without the SAP based NLM and the original NAMF with the SAP based NLM. The former one uses a traditional denoising mechanism, and the latter one adopts the SAP based NLM to further repair the noisy images. Figure 7 shows that the performances of the NAMF without the SAP based NLM are significantly weakened compared with the original NAMF.

FIGURE 4: Experimental results of different methods for "Lena" with SAP noise ratio of 90%. (a) Noisy image. (b) AMF. (c) NAFSMF. (d) AWMF. (e) [14]. (f) BPDF. (g) DAMF. (h) AFMF. (i) ASWMF. (j) NAMF. (k) Original image.

FIGURE 5: Continued.

FIGURE 5: Comparison of all methods in terms of PSNR, SSIM, MAE, and IEF at all SAP noise. (a) Average PSNR of different methods at all SAP noise levels. (b) Average SSIM of different methods at all SAP noise levels. (c) Average MAE of different methods at all SAP noise levels. (d) Average IEF of different methods at all SAP noise levels.

FIGURE 6: Average running time of different methods at all SAP noise levels.

FIGURE 7: Experimental results of the original NAMF and the NAMF without the SAP based NLM. (a) PSNR of NAMF and NAMF without the SAP based NLM. (b) SSIM of NAMF and NAMF without the SAP based NLM. (c) MAE of NAMF and NAMF without the SAP based NLM. (d) IEF of NAMF and NAMF without the SAP based NLM.

5. Conclusion

In this paper, a method called NAMF for SAP noise denoising is proposed, which adopts a SAP noise based nonlocal mean method. NAMF can get much higher restoring quality than state-of-the-art methods at all SAP noise levels. The processing time of NAMF is comparable to most state-of-the-art methods. The experimental results show that NAMF can get much better PSNR and SSIM at all SAP noise levels. Moreover, NAMF can preserve more details even at a noise level as high as 90%.

Data Availability

The data and codes in this study can be accessed at <https://github.com/zhuyuan-cug/NAMF>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the 111 Project under Grant B17040.

References

- [1] V. Chandra, S. Deokar, S. Badhe, and R. Yawle, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed adaptive median filter," *IEEE Signal Processing Letters*, vol. 18, no. 5, pp. 287–290, 2011.

- [2] W. Luo, "Efficient removal of impulse noise from digital images," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, pp. 523–527, 2006.
- [3] H. Fan, C. Li, Y. Guo, G. Kuang, and J. Ma, "Spatial-spectral total variation regularized low-rank tensor decomposition for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 6196–6213, 2018.
- [4] S. Xu, X. Yang, and S. Jiang, "A fast nonlocally centralized sparse representation algorithm for image denoising," *Signal Processing*, vol. 131, pp. 99–112, 2017.
- [5] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, "Image denoising review: from classical to state-of-the-art approaches," *Information Fusion*, vol. 55, pp. 220–244, 2020.
- [6] X. Guo, Y. Li, J. Ma, and H. Ling, "Mutually guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, pp. 694–707, 2020.
- [7] V. Singh, R. Dev, N. K. Dhar, P. Agrawal, and N. K. Verma, "Adaptive type-2 fuzzy approach for filtering salt and pepper noise in grayscale images," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 3170–3176, 2018.
- [8] B. Karthik, T. K. Kumar, S. Vijayaragavan, and M. Sriram, "Removal of high density salt and pepper noise in color image through modified cascaded filter," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–8, 2020.
- [9] H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results," *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 499–502, 1995.
- [10] R. H. Chan, H. Chung-Wa, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, 2005.
- [11] X. Deng, Y. Ma, and M. Dong, "A new adaptive filtering method for removing salt and pepper noise based on multilayered pcnn," *Pattern Recognition Letters*, vol. 79, pp. 8–17, 2016.
- [12] K. K. V. Toh and N. A. M. Isa, "Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 281–284, 2010.
- [13] P. Zhang and F. Li, "A new adaptive weighted mean filter for removing salt-and-pepper noise," *IEEE Signal Processing Letters*, vol. 21, no. 10, pp. 1280–1283, 2014.
- [14] W. Yi, J. Wang, S. Xiao, and H. Liang, "An efficient adaptive fuzzy switching weighted mean filter for salt- and-pepper noise removal," *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1582–1586, 2016.
- [15] U. Erkan and L. Gökrem, "A new method based on pixel density in salt and pepper noise removal," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 26, pp. 162–171, 2018.
- [16] U. Erkan, L. Gökrem, and S. Enginoğlu, "Different applied median filter in salt and pepper noise," *Computers & Electrical Engineering*, vol. 70, pp. 789–798, 2018.
- [17] U. Erkan, S. Enginoğlu, D. Thanh, and H. Le Minh, "Adaptive frequency median filter for the salt and pepper denoising problem," *IET Image Processing*, vol. 14, pp. 1291–1302, 05 2020.
- [18] D. N. H. Thanh, N. N. Hien, P. Kalavathi, and V. B. S. Prasath, "Adaptive switching weight mean filter for salt and pepper image denoising," *Procedia Computer Science*, vol. 171, pp. 292–301, 2020.
- [19] N. Asuni and A. Giachetti, "Testimages: a large-scale archive for testing visual devices and basic image processing algorithms," In *Eurographics Italian Chapter Conference*, vol. 1, p. 3, 2014.
- [20] L. P. Yaroslavsky, "Digital picture processing: an introduction," *Science & Business Media*, vol. 9, 2012.
- [21] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, pp. 839–846, IEEE, Washington, DC, 1998.
- [22] A. Buades, B. Coll, and J. . Morel, "A nonlocal algorithm for image denoising," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 60–65, San Diego, CA, USA, June 2005.
- [23] P. Coupe, P. Hellier, C. Kervrann, and C. Barillot, "Bayesian non local means-based speckle filtering," in *Proceedings of the 2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1291–1294, Paris, France, May 2008.
- [24] J. Froment, "Parameter-free fast pixelwise nonlocal means denoising," *Image Processing On Line*, vol. 4, pp. 300–326, 2014.
- [25] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, pp. 2366–2369, Istanbul, Turkey, August 2010.
- [26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [27] P. N. T. Wells, "Handbook of image and video processing," *Physiological Measurement*, vol. 22, no. 1, pp. 263–264, 2001.
- [28] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. Premchand, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," *IEEE Signal Processing Letters*, vol. 18, no. 5, pp. 287–290, 2011.