NAMF: A NON-LOCAL ADAPTIVE MEAN FILTER FOR SALT-AND-PEPPER NOISE REMOVAL

Houwang Zhang[1,2]　　*Yuan Zhu*[1,2,*]　　*Hanying Zheng*[1,2]

[1] China University of Geosciences, School of Automation, Wuhan, China
[2] Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan, China
[*] Corresponding author: Yuan Zhu (e-mail: zhuyuan@cug.edu.cn)

ABSTRACT

In this paper, a novel algorithm called a non-local 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 will be replaced by the combination of its neighboring pixels, and finally we use a SAP noise based non-local mean filter 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.

Index Terms— Adaptive mean filter, image denoise, non-local mean method, salt-and-pepper noise.

1. INTRODUCTION

Digital images are often corrupted by noises in the process of image acquisition and transmission [1, 2]. While the existence of noises will make tasks of image processing and computer vision become seriously ill-posed problems [3]. As a pre-processing step in image processing, image denoising can protect edges, textures and other details [4]. Hence it is taken as one of the most important tasks in image processing [5, 6]. Salt-and-pepper (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].

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 window with adaptive size, which makes it perform well in high noise intensity [11].

In recent years, studies on denoising of SAP noise are mainly based on MF and AMF. Some researches also use deep learning methods. However, these deep learning methods depend on data [12]. 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 [13]. Adaptive weighted mean filter (AWMF) uses two successive windows to detect noisy pixels and processes them with a weighted mean filter [14]. The method proposed in [15] is based on NAFSMF and AWMF and uses a new adaptive fuzzy switching weighted mean filter to remove SAP noise. In [16], 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 [17].

The restoration of SAP noise is just to use the rest information (uncontaminated pixels) to repair the absent information (contaminated pixels). As Fig. 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.
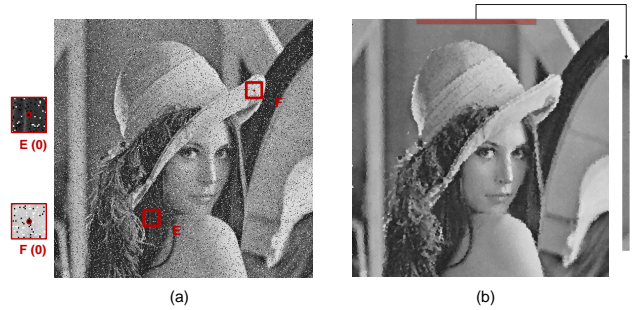
Fig. 1. (a) "lena" corrupted by SAP noise ratio of 10%, and the gray values of noisy candidates in E and F are both 0. (b) Result of the method proposed in [15] for "Lena" corrupted by SAP noise ratio of 90%.

To solve the above problem, in this paper, we propose a non-local adaptive mean filter (NAMF) to remove SAP noise efficiently. 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 value 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. Then we use a SAP noise based non-local mean filter to further restore it. The main contributions of this paper can be concluded as follows:

- A proportion based method is proposed to distinguish the noise pixels and texture pixels in the SAP noise detection, which can avoid noiseless texture pixels being processed as noisy pixels.

- An improved non-local mean method based on the characteristics of SAP noise is raised to further restore noisy images, which can further enhance the quality of restored images.

Experimental results on 16 typical images and 40 test images in the TEST IMAGES Database [18] demonstrate that NAMF outperforms the existing state-of-the-art methods under both high SAP noise level and low SAP noise level [1].

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 Non-local 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 X be an original 8-bit gray level image with size of $M \times N$, and $x_{i,j}$ be the gray value of the pixel at location (i, j). Then, for MF with filter size of 3×3, the new gray value $\tilde{x}_{i,j}$ of pixel at location (i, j) can be calculated by Eq. (1).

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

Based on the idea of MF, many MF methods have been proposed. However, when the distance between the noisy

---

pixel and its neighborhood is large, the difference of the median value pixel and the original pixel that is noiseless may be large, and that would cause the denoised image to be degraded [15]. In [14], the median value of neighborhoods is combined with different weights. For different neighborhoods, weights are computed based on 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 [15], to tackle the problem, the researchers provide a method by using neighborhoods in the restored image to calculate the median value.

2.2. Non-local Mean (NLM) Method

The NLM method combined the idea of the yaroslavsky filter [19] and bilateral filter [20]. For the noisy pixel, NLM use the intensity patch feature instead of single pixel feature to compute the denoised mean value [21].

Given a pixel $x \in I$, 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 Eqs. (2) - (4).

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

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

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

where $\tilde{x}$ is the restored gray value of x. $u(x, y)$ denotes the similarity between pixels x and y, which depends on the weighted Euclidean distance between two image blocks $N(x)$ and $N(y)$. h is the smoothing parameter, a is the standard deviation of the Gaussian kernel.

3. THE PROPOSED NON-LOACL ADAPTIVE MEAN (NAMF) FILTER

In this section we will present the details of our proposed non-local adaptive mean (NAMF) filter. As the most traditional SAP noise removal filters, NAMF can be 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 location (i, j) of X and Y, respectively, where $(i, j) \in \Lambda \equiv \{1, \cdots, M\} \times \{1, \cdots, N\}$.

In a corrupted image, the value of a "salt" pixel equals to the maximum gray value 255, and the value of a "pepper"

pixel equals to the minimum value 0. Thus, $y_{i,j}$ is defined by Eq. (5).

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

where α is the density of SAP noise of $\boldsymbol{Y}$. 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.

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

where $\boldsymbol{O} = (o_{i,j}) \in \mathbb{R}^{M \times N}$ 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 Fig. 1 (a), E and F are both candidates with 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 a 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.

- **Step 1** : For the pixel $y_{i,j}$ with $o_{i,j} = 1$, we calculate $S_{i,j}^{sum}(w)$ within an adaptive searching window $S_{i,j}(w)$. The radius w of window $S_{i,j}(w)$ is initialed to 1. If the condition is met ($w == w_{max}$ or $S_{i,j}^{sum}(w) > 0$), the w is just we need. If not, then $w+1$ and continue to compute, where $S_{i,j}^{sum}(w)$ 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 $S_{i,j}^{sum}(w) > 0$, we take the pixel $y_{i,j}$ as a noisy pixel, the detection for $y_{i,j}$ ends. Otherwise we go to **Step 2**.

- **Step 2** : If $S_{i,j}^{sum}(w) = 0$, $y_{i,j}$ maybe considered as a texture pixel. Then we calculate the proportion $\rho = \frac{S_{i,j}^{num}(w)}{(2w+1)\times(2w+1)}$, where $S_{i,j}^{num}(w)$ is the number of pixels owning same value as 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 $\rho \leq T$, the candidate pixel $y_{i,j}$ is regards 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 $\boldsymbol{L} = (l_{i,j}) \in \mathbb{R}^{M \times N}$, 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 $\boldsymbol{Z}$ represent the initially restored image, $\hat{\boldsymbol{Z}}$ represent the final output image. Before processing, we initialize $\boldsymbol{Z}$ with $\boldsymbol{Z} = \boldsymbol{Y}$.

Firstly, when a pixel $y_{i,j}$ is detected as a noisy pixel, we use $S_{i,j}^{mean}(w)$ to restore it. The calculation of $S_{i,j}^{mean}(w)$ is based on $S_{i,j}^{sum}(w)$ as shown in Eq. (7). When $S_{i,j}^{sum}(w) \neq 0$, $S_{i,j}^{mean}(w)$ is the mean of the noiseless pixels in $S_{i,j}(w)$, otherwise is the mean of three processed neighboring pixels of $y_{i,j}$ in $\boldsymbol{Z}$. Unlike the four neighbors adopted in [15], the utilization of four neighboring pixels will lead to residual noisy pixels on the boundary, as shown in Fig. 1. (b) (even if the image boundary is expanded during process, some noisy pixels located in boundary can still not be restored). Hence, we select three processed neighboring pixels in $\boldsymbol{Z}$, i.e., $S_{i,j}^{mean}(w) = (z_{i-1,j-1} + z_{i-1,j} + z_{i,j-1})/3$, and its illustration is shown in Fig. 2.

$$S_{i,j}^{mean}(w) = \begin{cases} \frac{\sum_{(e,f) \in S_{i,j}(w)} (1-l(e,f))*y_{e,f}}{\sum_{(e,f) \in S_{i,j}(w)} (1-l(e,f))}, & S_{i,j}^{sum}(w) \neq 0 \\ \frac{z_{i-1,j-1}+z_{i-1,j}+z_{i,j-1}}{3}, & otherwise \end{cases} \quad (7)$$
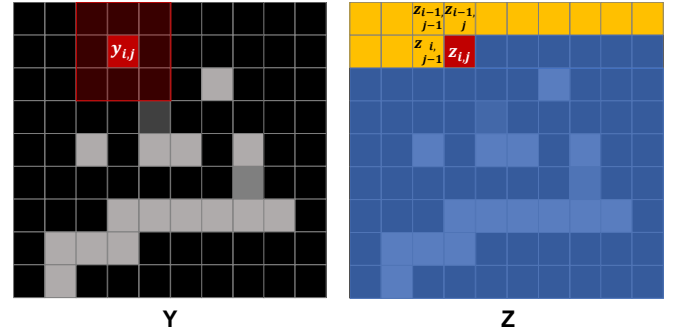
Fig. 2. $\boldsymbol{Y}$ is a noisy image, $\boldsymbol{Z}$ is an image being restored, the red pixel in $\boldsymbol{Z}$ is prepared to process, the yellow pixels are pixels which have been restored, other blue pixels in $\boldsymbol{Z}$ are pixels remained to be restored. When processing noisy pixel $y_{i,j}$, neighborhoods in the window are all noisy, i.e. $S_{i,j}^{sum}(w) = 0$. So here we select three neighboring pixels $z_{i-1,j-1}$, $z_{i-1,j}$ and $z_{i,j-1}$ in restored image $\boldsymbol{Z}$ for restoring pixel $y_{i,j}$ in original image $\boldsymbol{Y}$.

Secondly, considering the problem that restoration will be hard under high noise intensity, we introduce the non-local mean (NLM) method [21, 22] 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 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, pixel is identified and marked by the discriminant matrix $\boldsymbol{L}$. $l(i, j) = 1$ 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:

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

$$C = \sum_{z_{e,f} \in B(z_{i,j},\, r)} u(z_{i,j}, z_{e,f}), \tag{9}$$

$$, u(z_{i,j}, z_{e,f}) = \begin{cases} e^{-\frac{d(z_{i,j}, z_{e,f})}{h^2}}, & e \neq i \text{ or } f \neq j \\ 0, & e = i \text{ and } f = j \end{cases}$$

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

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 Eq. (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, h is the smoothing parameter for NLM.

When processing the noisy pixel $z_{i,j}$, 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 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 Eq. (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 Eq. (10), $\bar{L}$ represents the total number of non-zero elements in the discriminant matrix $\boldsymbol{L}$, 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 NLM algorithm, here we introduce a kind of fast implementation of NLM algorithm [23] 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.

Algorithm 1 NAMF

/*STAGE 1*/
  Compute the indicator matrix $\boldsymbol{O}$.
  For each pixel $(i, j) \in \Lambda$ in the noisy image $\boldsymbol{Y}$ and the initially restored image $\boldsymbol{Z}$, 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 $S_{i,j}^{sum}(w)$ until $w == w_{max}$ or $S_{i,j}^{sum}(w) > 0$;
   Otherwise, $w = w + h$ and repeat step 3).
4) If $S_{i,j}^{sum}(w) > 0$, $l(i, j) = 1$, $z_{i,j} = S_{i,j}^{mean}(w)$, break;
   Otherwise, go to step 5).
5) Compute ρ. If $\rho \leq T$, $l(i, j) = 1$, $z_{i,j} = S_{i,j}^{mean}(w)$;
   Otherwise, $l(i, j) = 0$, $z_{i,j} = y_{i,j}$.

/*STAGE 2*/
  Compute h.
  For each pixel $z_{i,j}$ in image $\boldsymbol{Z}$ and $\hat{z}_{i,j}$ in output image $\hat{\boldsymbol{Z}}$, do
6) If $l(i, j) == 1$, $\hat{z}_{i,j} = \frac{1}{C} \sum_{z_{e,f} \in B(z_{i,j},\, r)} z_{e,f} * u(z_{i,j}, z_{e,f})$;
   Otherwise, $\hat{z}_{i,j} = z_{i,j}$.

---

4. EXPERIMENTAL RESULTS

In the experiments, NAMF are compared with six state-of-the-art methods: AMF [9], NAFSMF [13], AWMF [14], the method proposed in [15], BPDF [16], and DAMF [17]. 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 [18] are chosen for the experiments.

We use two typical image quality metrics, peak signal-to-noise ratio (PSNR) [24] and structural similarity (SSIM) [25] to evaluate the experimental results. For an image U and an image V with same size of $M \times N$, PSNR can be calculated as follows:

$$PSNR(U, V) = 10 * log_{10}(255^2 / MSE),$$
$$MSE = \frac{1}{M * N} \sum_i^M \sum_j^N (U_{i,j} - V_{i,j})^2 \tag{11}$$

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

Fig. 3. Experimental results of different methods for "Barbara" with SAP noise ratio of 10%. (a) Noisy image (15.4199 dB, 0.2586). (b) AMF (24.4698 dB, 0.7744) (c) NAFSMF (33.6841 dB, 0.9788). (d) AWMF (32.1548 dB, 0.9644). (e) [15] (34.7489 dB, 0.9820). (f) BPDF (32.6820 dB, 0.9720). (g) DAMF (33.6841 dB, 0.9788). (h) NAMF (**41.3133 dB**, **0.9932**). (i) Original image.
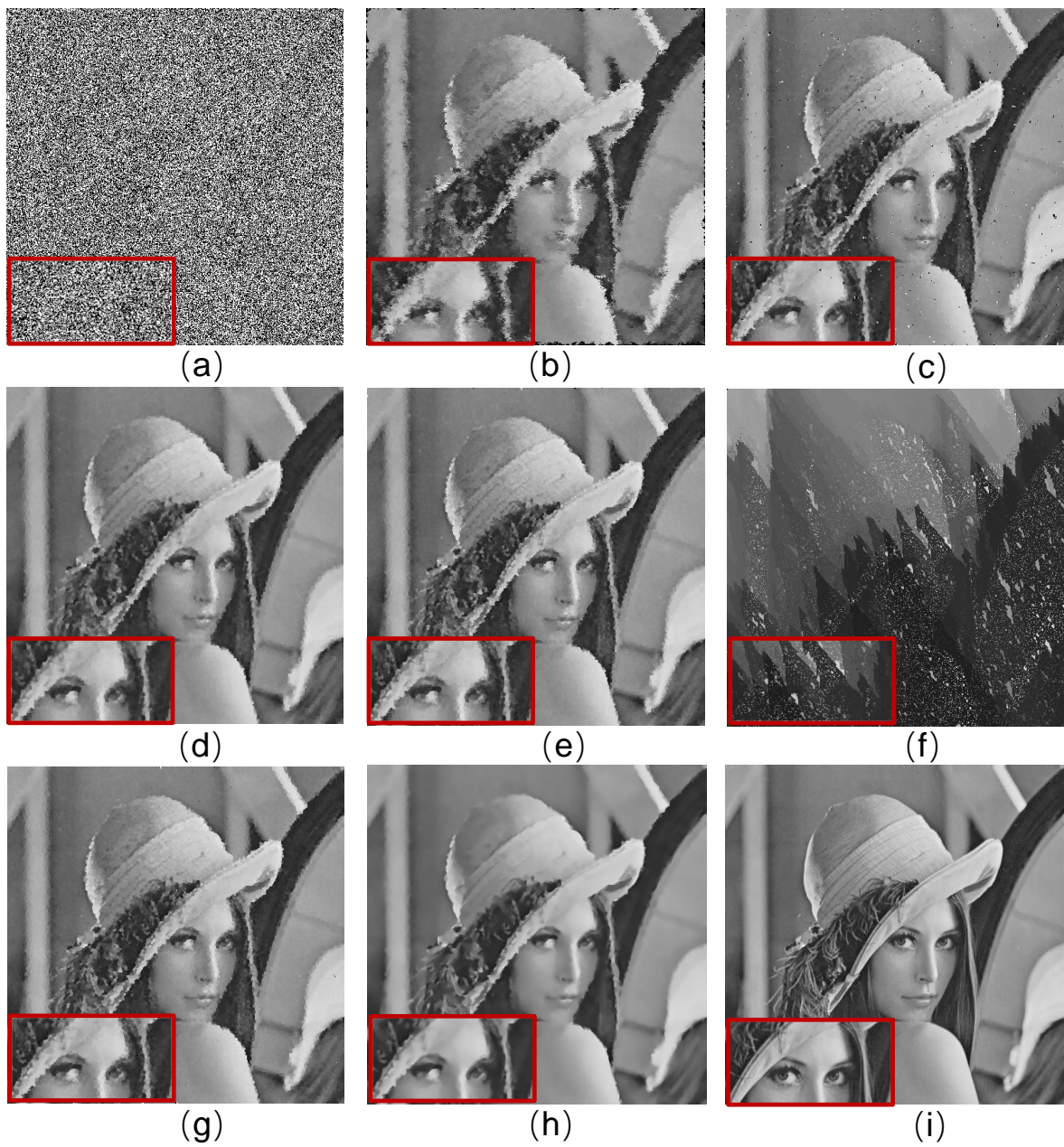
Fig. 4. Experimental results of different methods for "Lena" with SAP noise ratio of 90%. (a) Noisy image (5.8973 dB, 0.006). (b) AMF (20.0591 dB, 0.5860) (c) NAFSMF (23.7711 dB, 0.6959). (d) AWMF (26.1224 dB, 0.7652). (e) [15] (26.1986 dB, 0.7710). (f) BPDF (10.8694 dB, 0.2775). (g) DAMF (25.9059 dB, 0.7631). (h) NAMF (**27.5748 dB**, **0.8150**). (i) Original image.
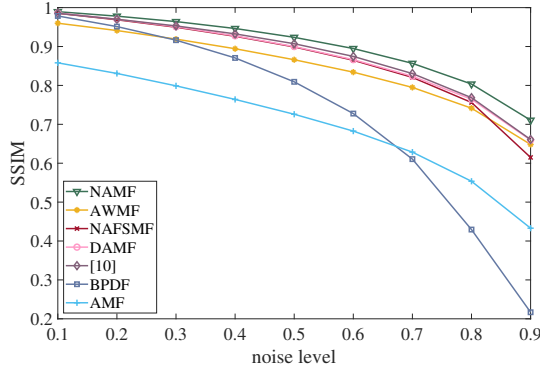
image U and V can be defined as follows:

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

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 [25], respectively.
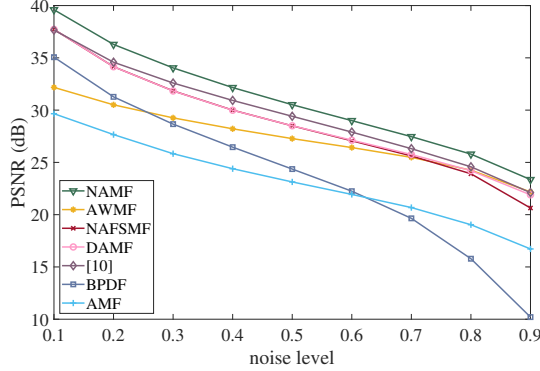
Fig. 5. Average PSNR of different methods at all SAP noise levels.

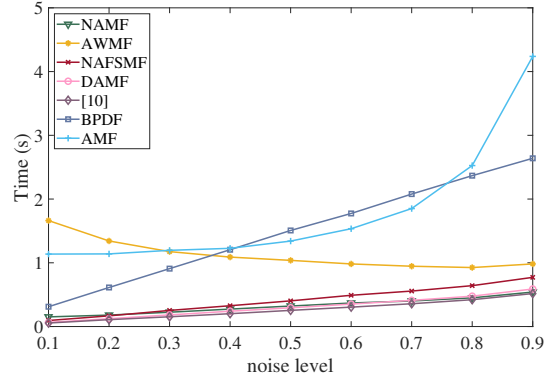**Fig. 6**. Average SSIM of different methods at all SAP noise levels.

In this paper, we set $\beta_2 = 2.2186$, $\beta_1 = 6.0314$ and $\beta_0 = 4.5595$ to fit h. The size of searching window is 5×5. And the size of similarity window is 41×41. Through test, we finally take 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.

Fig. 3 shows the restored results of all methods for "Barbara" with SAP noise ratio of 10%. By observing the enlarged area in Fig. 3, we can see that AMF, AWMF, and BPDF can't restore the details of the original image very well. And the re-

sult of our proposed NAMF is nearly the same as the original image.

Fig. 4 shows the restored results for "Lena" with SAP noise ratio of 90%. In the enlarged area in Fig. 4, it can be found that the performances of AMF, NAFSMF, and BPDF are very poor. And the restored images by other methods are also very blurred, while result of our method looks more natural and smooth.

The curves of average PSNR and SSIM are shown in Fig. 5 and Fig. 6 respectively. Fig. 5 illustrates that NAMF obtains the highest PSNR under both low and high noise intensity, and PSNR of our method is much higher than results of other methods. Fig. 6 shows that the SSIM curves of most methods are basically the same under low SAP noise intensity. However, with the increasing of noise level, the superiority of NAMF is getting more obviously. After the noise ratio exceeding 30%, the SSIM obtained by NAMF is significantly higher than other methods.

Fig. 7. Average running time of different methods at all SAP noise levels.

Fig. 7 illustrates the average running time of different methods at all noise levels. With the increasing of noise intensity, the running time of all methods except AWMF increases. Clearly, the average running time of the method proposed in [15] is the shortest. Although the rank of NAMF is in the middle, its processing speed is superior to NAFSMF and similar with DAMF under high noise intensity.

5. CONCLUSION

In this paper, a method called NAMF for SAP noise denoising is proposed, which adopts a SAP noise based non-local 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 noise level as high as 90%.

6. REFERENCES

[1] Vivek Chandra, Sagar Deokar, Siddhant Badhe, and Rajesh 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] Wenbin Luo, "Efficient removal of impulse noise from digital images," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, pp. 523–527, 2006.

[3] Haiyan Fan, Chang Li, Yulan Guo, Gangyao Kuang, and Jiayi 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] Xiaojie Guo, Yu Li, Jiayi Ma, and Haibin Ling, "Mutually guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, pp. 694–707, 2020.

[5] Shaoping Xu, Xiaohui Yang, and Shunliang Jiang, "A fast nonlocally centralized sparse representation algorithm for image denoising," *Signal Processing*, vol. 131, pp. 99–112, 2017.

[6] Bhawna Goyal, Ayush Dogra, Sunil Agrawal, BS Sohi, and Apoorav Sharma, "Image denoising review: From classical to state-of-the-art approaches," *Information Fusion*, vol. 55, pp. 220–244, 2020.

[7] Vikas Singh, Raghav Dev, Narendra K Dhar, Pooja Agrawal, and Nishchal 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 Krishna Kumar, SP 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] Raymond H Chan, Ho Chung-Wa, and Nikolova Mila, "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] Xiangyu Deng, Yide Ma, and Min 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] Yuesheng He and Lionel M Ni, "A novel scheme based on the diffusion to edge detection," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1613–1624, 2018.

[13] Kenny Kal Vin Toh and Nor Ashidi Mat 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.

[14] Peixuan Zhang and Li Fang, "A new adaptive weighted mean filter for removing salt-and-pepper noise," *IEEE Signal Processing Letters*, vol. 21, no. 10, pp. 1280–1283, 2014.

[15] Wang Yi, Jiangyun Wang, Song Xiao, and Han 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.

[16] Uğur Erkan and Levent Gökrem, "A new method based on pixel density in salt and pepper noise removal," *Turkish Journal of Electrical Engineering and Computer Science*, vol. 26, pp. 162–171, 2018.

[17] Uğur Erkan, Levent Gökrem, and Serdar Enginoğlu, "Different applied median filter in salt and pepper noise," *Computers & Electrical Engineering*, vol. 70, pp. 789–798, 2018.

[18] Nicola Asuni and Andrea Giachetti, "Testimages: a large-scale archive for testing visual devices and basic image processing algorithms.," in *Eurographics Italian Chapter Conference*, 2014, vol. 1, p. 3.

[19] Leonid P Yaroslavsky, *Digital picture processing: an introduction*, vol. 9, Springer Science & Business Media, 2012.

[20] Carlo Tomasi and Roberto Manduchi, "Bilateral filtering for gray and color images," in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998, pp. 839–846.

[21] A. Buades, B. Coll, and J. . Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005, vol. 2, pp. 60–65 vol. 2.

[22] Pierrick Coupe, Pierre Hellier, Charles Kervrann, and Christian Barillot, "Bayesian non local means-based speckle filtering," pp. 1291–1294, 2008.

[23] Jacques Froment, "Parameter-free fast pixelwise non-local means denoising," *Image Processing On Line*, vol. 4, pp. 300–326, 2014.

[24] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *2010 20th International Conference on Pattern Recognition*, Aug 2010, pp. 2366–2369.

[25] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.