International Journal of Innovative Computing, Information and Control
Volume **13**, Number **4**, August **2017**

ICIC International ©2017 ISSN 1349-4198
pp. 1061–1075

A MODIFIED PM DIFFUSION METHOD FOR SALT-AND-PEPPER NOISE REMOVAL

Hongyao Deng[1,3,*], Qingxin Zhu[1] and Xiuli Song[2]

[1]School of Information and Software Engineering
University of Electronic Science and Technology of China
No. 2006, Xiyuan Ave., West Hi-Tech Zone, Chengdu 611731, P. R. China
[*]Corresponding author: HYDeng_2004@163.com; qxzhu@uestc.edu.cn

[2]School of Computer Science and Technology
Chongqing University of Posts and Telecommunications
No. 2, Chongwen Rd., Nan-an District, Chongqing 400065, P. R. China
songxl@cqupt.edu.cn

[3]College of Computer Engineering
Yangtze Normal University
No. 98, Julong Rd., Lidu Fuling District, Chongqing 408000, P. R. China
denghongyao@yznu.cn

Received December 2016; revised April 2017

Abstract. *Reducing salt-and-pepper noise usually employs median filtering, switching median filtering, the total variation ℓ_1 and variants. These approaches, however, often introduce excessive smoothing and can result in extensive visual feature blurring and thus are suitable only for images with low noise. A new method to suppress noise is proposed in this paper that overcomes this limitation that modifies PM method. In contrast to the original PM method proposed by Perona and Malik, the modified PM method only treats noisy pixels, and diffuses along eight-neighbors directions, rather than all pixels and along four-neighbors directions. Therefore, the modified PM method is suitable for images not only with low noise but also with high noise, and can reconstruct more details from noisy images. In addition, a selected mean filter is also proposed to obtain an initial estimate of original noise-free image. The initial estimate is taken as the initialization image for iteratively diffusion process, rather than original noisy image, and thus the computational cost is reduced significantly. Experimental results show that the new method has strong capability to reduce noise in terms of its robustness and very good denoising results.*
Keywords: Gray-scale image, Nonlinear diffusion, Noise detection, Noise removal

1. **Introduction.** Images are often corrupted by salt-and-pepper noise, which may arise due to malfunctioning camera photo-sensors, optic imperfections, or transmission errors during the acquisition. Corrupted pixel presents itself as sparsely occurring white and black points. Noise suppression is of great benefit in many applications such as image segmentation, object identification and image fusion. Therefore, rounding the problem of how to yield better images from their noisy versions, a series of various methods have been presented.

Simple and direct methods are generalized median filtering and related nonlinear filtering techniques [1], such as median (MED) filter [2], weighted median filter (WMF) [3], center weighted median filter (CWMF) [4], adaptive median filter (AMF) [5] and adaptive center-weighted median filter (ACWMF) [6]. These filters show differences between one another although they output an observed pixel similarly by the median pixel of its neighborhood, also termed sliding window. The MED filter uses a fixedly sized sliding window. WMF and CWMF apply a weight adjustment by duplicating the pixels within a sliding window. AMF adopts dynamic sliding windows whose sizes depend on local noise density. ACWMF incorporates the AMF and CWMF techniques. These methods treat corrupted and uncorrupted pixels uniformly and result in too much smoothing to different degrees, and fail to process images with high noise.

A family of switching filtering has been presented to overcome these drawbacks that makes a distinction between noisy and noise-free pixels, and thus usually contains noise detection module and noise removal module. The former identifies corrupted pixels, and the latter applies a technique to these corrupted pixels, while uncorrupted pixels are left unchanged. From the switching filtering structure, noise detection is crucial. If a detection module fails to identify corrupted pixels, then these noisy pixels will be left unchanged and result in poor filtered image; if the detection module classifies corrupted pixels correctly, but also declares the noise-free pixels as noisy, many of the image details will be lost. Rounding how to devise a good detector, many techniques have been proposed.

To the best of our knowledge, the presented detection techniques are roughly grouped into three classifications. The first technique compares the noisy image with an estimate of its noise-free image pixel-by-pixel. If the absolute difference of intensities of paired pixels is larger than a predefined threshold, the corresponding pixel is declared corrupted; otherwise, it is declared as noise-free. There are many methods to obtain the estimate. For example, a median based impulse detector is devised in [7]; a weighted median filter is employed in [8]; a non-local median filter is used in [9]. By exploiting the local neighborhood, the second technique devises a metric to judge a pixel is corrupted or not. For example, the rank-ordered absolute differences (ROAD) is used in [10]; the first eight minimum aggregated intensities within a sliding window that contains twenty-five pixels are used as a statistic in [11]. And further, the local neighborhood statistic with directions is also exploited, such as the four-directions statistic in [12], the twelve-directions statistic in [13], and the multi-phase statistic in [9, 14]. The third technique employs artificial intelligence techniques, such as artificial neural network (ANN) [15], neuro-fuzzy network (NFN) [16], and support vector machine (SVM) [17]. Of course, there are some methods containing multiple detective techniques such as [18].

Apart from median filtering and switching filtering, partial differential equations use a variational energy minimization of an objective function to obtain restoration images. The objective function usually consists of a data fidelity term and a regularization term. The former depends on different noise models; the latter is an image prior. For salt-and-pepper noise model, the data fidelity term is the ℓ_1 norm of residual error, derived from the maximum a posterior (MAP) probability estimate. Image prior is an assumption for image characteristics. Therefore, this approach involves what the prior is and how to optimize an objective function. For example, a sparse representation is taken as the prior, and a dictionary learning method is employed to optimize the objective function in [19]; total variation is taken as the prior, and the steepest-descent method is used to optimize the objective function in [20]; ℓ_0 norm is taken as the prior, and the primal-dual algorithm is utilized to optimize the objective function in [21].

Inspired by switching filtering and variational methods, a modified PM diffusion method is proposed to reduce salt-and-pepper noise in this paper. Similar to switching filtering, the proposed method consists of detection and removal modules. In the detection module, an efficient noise detector identifies noisy pixels. In the removal module, the modified PM diffusion operations are iteratively implemented on corrupted pixels until a terminal condition is satisfied, while uncorrupted pixels are left unchanged. Experimental results show that the proposed method is effective for salt-and-pepper removal in terms of its robustness and very good denoising results.

A new method is proposed for salt-and-pepper noise removal in this paper. The four main contributions are as follows.

- A modified PM diffusion method is proposed for salt-and-pepper noise removal. In contrast to original PM method, there are two crucial differences. Firstly, the modified method only treats noisy pixels in an image domain, rather than all pixels, and thus is suitable for images not only with low noise but also with high noise. Secondly, the modified PM method implements diffusion operations along eight-neighbors directions, rather than along four-neighbors directions, and thus can reconstruct more details from noisy images.
- A simple and effective selected mean filter is also proposed in order to obtain an evolved intermediate image. This filter takes the arithmetical mean of uncorrupted pixels within a sliding window as a pixel output, based on the fact that in noise-free images, all the pixels within a sliding window are similar to the current pixel centered in the sliding window.
- An evolved intermediate image is taken as the initialization image in iterative diffusion process, rather than original noisy image. And thus the computational cost of the proposed method is decreased significantly. The evolved intermediate image is obtained by the selected mean filter mentioned above.
- The proposed method has strong capability to remove salt-and-pepper noise in terms of the robustness to different density noises and the very good denoising results, as seen in PSNR/SSIM evaluations and visual quality of restored images.

The rest of the paper is organized as follows. In Section 2, a few existing diffusion methods are introduced. Section 3 describes the details of noise removal. Experimental results and comparisons are discussed in Section 4, and conclusions are drawn in Section 5.

2. **Existing Diffusion Methods.** Diffusion is an approach to image denoising. Different diffusion modes are introduced in this section. Let u be a gray-scale image, and c be a spatially-varying diffusion coefficient. In addition, t denotes a time step, and div and ∇ denote divergence and gradient operators, respectively. The diffusion framework is

$$u_t = div(c \cdot \nabla u). \tag{1}$$

Linear diffusion [22] is the simplest diffusion mode. Its diffusivity function is a constant, and is usually set to 1, i.e.,

$$c(x, y, t) = 1. \tag{2}$$

The corresponding diffusion equation is

$$\frac{\partial}{\partial t}u(x, y, t) = \Delta u(x, y, t) = \frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2}, \tag{3}$$

where Δ is a Laplacian operator. Solving the linear diffusion equation is equivalent to a linear filtering of signal u by convolution, i.e., $u(x, y, t) = u(x, y, 0) * G_{\sqrt{2t}}$, where $*$ denotes convolution operator and $G_{\sqrt{2t}}$ denotes a Gaussian skeleton with zero mean and variance of $\sqrt{2t}$. Linear diffusion has the drawback similar to Gaussian filtering. It introduces too much smoothing and results in blurred sharp features in filtered images.

PM diffusion proposed by Perona & Malik is a nonlinear diffusion [23]. The authors proposed two diffusivity functions as follows,

$$c_1(x, y, t) = \exp\left(-\left(\frac{|\nabla u(x, y, t)|}{\lambda}\right)^2\right) \tag{4}$$

and

$$c_2(x,y,t) = \left(1 + \left(\frac{|\nabla u(x,y,t)|}{\lambda}\right)^2\right)^{-1}, \tag{5}$$

where λ is referred to as the diffusion constant. They are the functions of signal gradients, and decrease monotonously. Since the conduction is only along edges, the PM diffusion method preserves edges and controls smoothing. However, PM diffusion method is sensitive to noise, and the corresponding diffusion equation is ill-posed [24].

The following diffusion equation [24] can effectively tackle the ill-posed problem of PM diffusion equation, given by

$$\frac{\partial}{\partial t}u(x,y,t) = div\left(c\left(|\nabla G_\sigma * u|\right)\nabla u\right), \tag{6}$$

where G_σ denotes a Gaussian smoothing kernel and σ is a given variance of the Gaussian smoothing kernel.

A Laplacian-based diffusivity function has been proposed by You and Kaveh [25], given by

$$c\left(|\Delta u|\right) = \left(1 + \left(\frac{|\Delta u(x,y,t)|}{\lambda}\right)^2\right)^{-1}. \tag{7}$$

In contrast to PM diffusivity function, the gradient magnitude is replaced by Laplacian magnitude, and thus the corresponding diffusion equation turns from the second-order to the fourth-order.

The diffusion equation proposed by Yu et al. [26] contains an edge detector, as follows

$$\frac{\partial}{\partial t}u(x,y,t) = div\left(SUSAN\left(c\left(|\nabla G_\sigma * u|\right)\right)\nabla u\right). \tag{8}$$

The SUSAN detector guides the diffusion process in an effective manner.

The diffusion equation in [27] is with weighted diffusivity function, given by

$$\frac{\partial}{\partial t}u(x,y,t) = div\left(w(x,y,t)c(x,y,t)\nabla u\right), \tag{9}$$

where $w(x,y,t)$ is a pixel-wise weight function.

In addition, nonlinear diffusion is employed in some transform domains in recent years, such as discrete wavelet domain [28] and shearlet domain [29]. Such work opens up a new way to utilize diffusion to remove impulsive noise.

3. The Proposed Method.

In this section, the details of the proposed method are introduced. In Section 3.1, the noise model and the denoising structure are described. Section 3.2 and 3.3 formulate the noise detection and diffusion details, respectively, and the description of acceleration is shown in Section 3.4.

3.1. Noise model and denoising structure.

Gray-scale images with salt-and-pepper noise are only considered in this paper. Let $u_{i,j}$ be a pixel at position (i,j), and a contamination image can be modeled as

$$u_{i,j} = \begin{cases} v_{i,j} & \text{with probability } \pi \\ o_{i,j} & \text{with probability } 1-\pi \end{cases}, \tag{10}$$

where $o_{i,j}$ denotes the noise-free pixel, and the corrupted pixel $v_{i,j}$ is a random variable which takes on the values 0 or 255, assuming 8-bit single channel image representation.

The denoising structure is shown in Figure 1. As seen in Figure 1, the denoising structure consists of a detection module and a removal module. The denoising module divides pixels into noisy and the noise-free pixels. Based on detection results, a mask

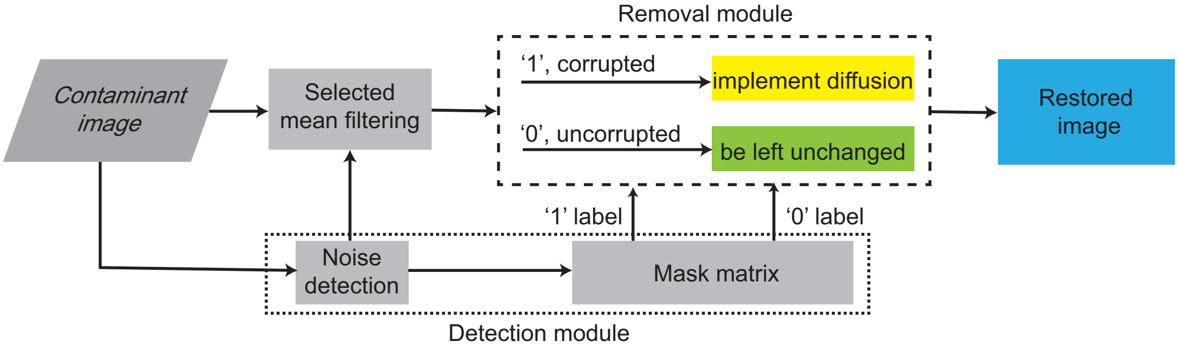


FIGURE 1. The noise reduction structure

matrix is built, in which each entry is a binary label indicating the corresponding pixel is corrupted or not. The removal module implements diffusion operations iteratively following the mask matrix instructions; as in Figure 1, the operations surrounded by the rectangle with dashed lines are repeated.

3.2. **Noise detection.** The devised detector is introduced in this subsection. Let $u_{x,y}$ be a current pixel, centered in the corresponding sliding window. The sliding window, also termed the neighborhood of the current pixel $u_{x,y}$, is defined by

$$N_{x,y}(R) = \{u_{i,j} : |x-i| \le R, |y-j| \le R\} \quad \text{for } (x,y) \in \Omega, \tag{11}$$

where the positive integer R and Ω denote neighborhood radius and image domain, respectively. Obviously, a neighborhood contains $(2R+1)*(2R+1)$ pixels.

From the noise model (10), only the pixels with intensity 0 or 255 are the noisy pixel candidates. Assuming the current pixel $u_{x,y}$ is a candidate, the absolute differences of intensities between it and every pixel within the neighborhood are calculated, and then the number of pixels is whose absolute difference is larger than a predefined threshold value determined, expressed as

$$m_{x,y} = \#\left\{u_{i,j} \in N_{x,y}(R) \ : \ |u_{x,y} - u_{i,j}| > T\right\}, \tag{12}$$

where # denotes the cardinality of the set, T is the predefined threshold value, and $m_{x,y}$ denotes the number of the pixels that satisfy the threshold condition.

The number $m_{x,y}$ is used to determine if candidate $u_{x,y}$ is corrupted, based on the following twofold, as a noise-free image usually consists of local smoothly varying areas separated by edges, while a salt-and-pepper noise takes a value substantially larger or smaller than its neighbors.

- If $u_{x,y}$ is a noisy pixel surrounded by a flat region, $m_{x,y}$ is very large with high probability; whereas, if $u_{x,y}$ is a noise-free pixel, $m_{x,y}$ is very small with high probability.
- If $u_{x,y}$ is a noisy pixel riding on edges, $m_{x,y}$ is large with high probability; whereas, if $u_{x,y}$ is a noise-free pixel, $m_{x,y}$ is small with high probability.

Therefore, $m_{x,y}$ can be used to determine $u_{x,y}$ is a noisy pixel or not by a given threshold, denoted by thr. The determining function is

$$f(x,y) = \begin{cases} 1, & \text{if } m_{x,y} > thr \\ 0, & \text{otherwise} \end{cases}. \tag{13}$$

In this equation, $f(x,y) = 1$, then candidate $u_{x,y}$ is corrupted; otherwise, it is uncorrupted. When all candidates are complete, a binary mask matrix can be built indicating the corresponding pixel is corrupted or not.

3.3. **Diffusion details.** If noise detection is complete, and corresponding mask matrix has been built, then modified PM diffusion operations are iteratively implemented on the corrupted pixels while the uncorrupted pixels are left unchanged. The modified PM diffusion is detailed in this subsection. Letting u_0 be a noisy image, and $\Theta \subset \Omega$ be the corrupted pixel domain, the modified PM diffusion is as follows

$$\begin{cases} \dfrac{\partial}{\partial t} u(x,y,t) = div\left(\dfrac{1}{1+(|\nabla u|/\lambda)^2}\cdot \nabla u\right), \text{ for } (x,y)\in\Theta \\ u(x,y,0) = u_0 \end{cases}. \tag{14}$$

In this formula, the parameter λ is referred to as the diffusion constant. In contrast to the original PM diffusion, the modified PM diffusion only treats the corrupted pixel domain Θ, rather than the image domain Ω.

The explicit discrete scheme for (14) must be discussed. Let D denote the first-order forward finite-difference operator. Based on the theory, $\nabla u \simeq Du$, for any pixel $u_{i,j}$ the eight formulas hold, as follows

$$D_N u_{i,j} = u_{i-1,j} - u_{i,j}, \quad D_E u_{i,j} = u_{i,j+1} - u_{i,j}, \quad D_{NE} u_{i,j} = u_{i-1,j+1} - u_{i,j},$$
$$D_{SE} u_{i,j} = u_{i+1,j+1} - u_{i,j}, \quad D_S u_{i,j} = u_{i+1,j} - u_{i,j}, \quad D_W u_{i,j} = u_{i,j-1} - u_{i,j},$$
$$D_{SW} u_{i,j} = u_{i+1,j-1} - u_{i,j}, \quad D_{NW} u_{i,j} = u_{i-1,j-1} - u_{i,j}.$$

In these eight equations, these subscripting symbols $\{N,E,S,W,NE,SE,SW,NW\}$ denote the eight diffusion directions, as seen in Figure 2. And assuming grid unit distances $dx = dy = 1$, thus the diagonal grid unit distance is $\sqrt{2}$. Then, the explicit discrete scheme for (14) can be written as

$$u_{i,j}^{n+1} = u_{i,j}^n + (\Delta t)\begin{bmatrix} c_N\left(D_N u_{i,j}^n\right)\cdot D_N u_{i,j}^n + c_E\left(D_E u_{i,j}^n\right)\cdot D_E u_{i,j}^n \\ + c_S\left(D_S u_{i,j}^n\right)\cdot D_S u_{i,j}^n + c_W\left(D_W u_{i,j}^n\right)\cdot D_W u_{i,j}^n \\ + \frac{1}{2}c_{NE}\left(D_{NE} u_{i,j}^n\right)\cdot D_{NE} u_{i,j}^n + \frac{1}{2}c_{SE}\left(D_{SE} u_{i,j}^n\right)\cdot D_{SE} u_{i,j}^n \\ + \frac{1}{2}c_{SW}\left(D_{SW} u_{i,j}^n\right)\cdot D_{SW} u_{i,j}^n + \frac{1}{2}c_{NW}\left(D_{NW} u_{i,j}^n\right)\cdot D_{NW} u_{i,j}^n \end{bmatrix}, \tag{15}$$

where Δt is the time step-size, the superscript n denotes the n-th iteration, and c with subscript denotes the diffusivity function along the subscript direction.

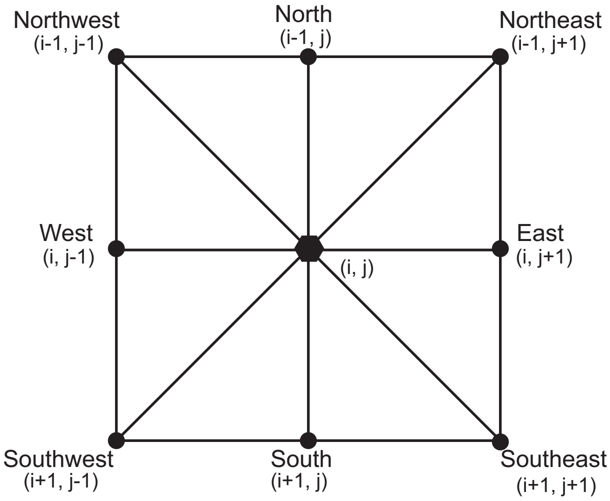


FIGURE 2. The current pixel and the eight diffusion directions

3.4. **Acceleration.** To decrease the number of iterations, the noisy image u_0 in (14) is replaced with an evolved intermediate image, denoted by $\hat{u}_0$. The evolved intermediate image can be obtained by a filter termed selected mean filter. The selected mean filter consists of noise detection and noise removal. The detector is the same as the devised in Section 3.2, and the removal module takes the arithmetical mean value of uncorrupted noise within the corresponding sliding window as the current output.

Assuming the noise detection is complete, thus the noise-free pixel set can be obtained, denoted by Θ^c, where the superscript c denotes complement operator. The noise-free neighborhood is defined by

$$N_{x,y}^0(x) = \{u_{i,j} : u_{i,j} \in N_{x,y}(r) \text{ and } (i,j) \in \Theta^c\}, \tag{16}$$

where $N_{x,y}(r)$ is the neighborhood of pixel $u_{x,y}$, with neighborhood radius r. Let symbol *mean* denote arithmetical mean operator, and the selected mean filter is expressed as

$$\hat{u}(x,y) = mean\left\{N_{x,y}^0(r)\right\}, \text{ if } (x,y) \in \Theta. \tag{17}$$

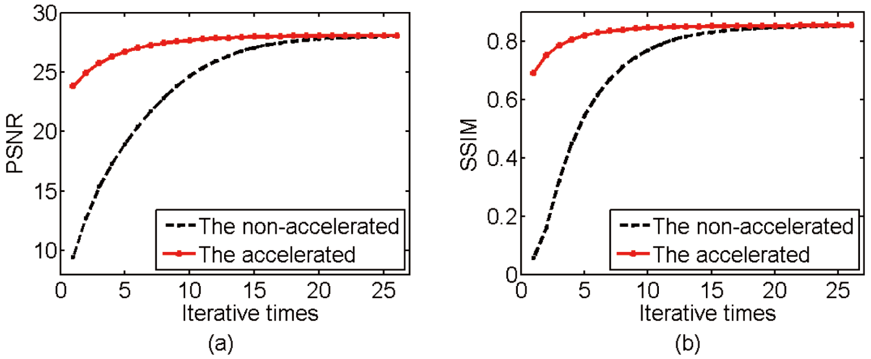
When all noisy pixels are processed, the evolved intermediate image $\hat{u}_0$ can be obtained. Thus, the accelerated diffusion equation can be written as

$$\begin{cases} \dfrac{\partial}{\partial t}u(x,y,t) = div\left(\dfrac{1}{1+(|\nabla u|/\lambda)^2}\cdot\nabla u\right), \text{ for } (x,y)\in\Theta \\ u(x,y,0) = \hat{u}_0 \end{cases}. \tag{18}$$

Next, the acceleration performance is validated. For ease of description, Equation (14) is called the non-accelerated, and Equation (18) is termed the accelerated. The two methods were applied to the same test image Barbara, sized 512 × 512 and with ten different levels noise. The parameter values $\{R = 2, T = 50, thr = 3, \lambda = 180, \Delta t = 1/7\}$ were used, and the parameter $r = 5$ was only used in the accelerated. The number of iterations was noted when PSNR reached the peak value during iterative processing, and the same action was made for SSIM measurements. The notions about PSNR and SSIM are introduced in Section 4.1. So, the number of iterations before and after acceleration was obtained, and reported in Table 1. To exhibit the acceleration process clearly, the

Table 1. The number of iterations before and after acceleration

π	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Before	5	6	8	10	12	18	20	27	48	50
After	1	2	3	3	4	4	6	9	10	11



(a)

(b)

Figure 3. Comparison of iteration progress between before and after acceleration

image Straw, sized 1024 × 1024 and with 70% density noise, was also tested. With the increasing of iterative times, the PSNR and SSIM values are plotted in Figure 3. From these test results, the accelerated significantly decreases the number of iterations, and the iteration number of the accelerated is about a fifth of the non-acceleration when time step-size is set to 1/7.

4. **Experiments and Comparisons.** In this section, two metrics used in performance evaluation are first introduced, then the experimental results are exhibited, and comparisons are finally discussed.

4.1. **Two metrics.** Peak signal-to-noise-ratio (PSNR) measurement is based on pixel intensity errors between noise-free and restored images. The calculation of PSNR is as follows

$$PSNR = 10\log_{10}\left(\frac{255^2}{|u|^{-1}\|u-\hat{u}\|_F^2}\right), \tag{19}$$

where $|\bullet|$ is the cardinality of an image, $\|\bullet\|_F$ denotes Frobenius norm, and u and $\hat{u}$ are the noise-free and the restored images, respectively.

Structural similarity index measure (SSIM) measurement is based on structural similarity. Its computation involves two blocks, denoted by y_1 and y_2. Let μ_{y_1}, μ_{y_2} be the mean values of y_1 and y_2, respectively, σ_{y_1} and σ_{y_2} be the variances, and $\sigma_{y_1y_2}$ be the covariance; thus the calculation of SSIM is as follows

$$SSIM(y_1,y_2) = \frac{(2\mu_{y_1}\mu_{y_2}+c_1)(2\sigma_{y_1y_2}+c_2)}{\left(\mu_{y_1}^2+\mu_{y_2}^2+c_1\right)\left(\sigma_{y_1}^2+\sigma_{y_2}^2+c_2\right)}, \tag{20}$$

where c_1 and c_2 denote two stabilization variables. Actually, this metric is the mean SSIM that gives the mean value of the structural similarity between the blocks of noise-free image and restored image. In this paper the SSIM is referred to as the mean SSIM.

4.2. **Setting parameters.** A total of seven parameters are set in the accelerated method, shown in Table 2. R, T and thr are used in the detection module; λ, Δt and N are used in the removal module; r is used in the selected mean filter. The parameter R is set to 2 when noise density $\pi < 40\%$; otherwise, it is set to 3. For all noise levels, T, thr and λ are set to 55, 3 and 180, respectively. The time step-size Δt is set to 1/7 that satisfies the CFL criteria. The parameter N depends on noise density π and time step-size Δt. The value of N is shown in Table 1 when $\Delta t = 1/7$. The parameter r depends on noise density. The value of it must be greater than or equal to the number of noisy pixels within a sliding window. The parameter value of r used in our experiments is shown in Table 3.

Table 2. The seven parameters used to the accelerated method

Para.	**Comments**
R	Neighborhood radius used in noise detection.
T	Threshold for judging the relationship between a candidate and its neighbors.
thr	Threshold that decides whether a candidate is corrupted or not.
λ	Diffusion constant.
Δt	Time step-size.
N	Desirable number of iterations.
r	Filtering radius used in the selected mean filter.

Table 3. The setting of parameter r on different density noise

π	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
r	1	1	2	3	4	4	5	5	6	6

4.3. **Experimental results.** A test set was built to evaluate the accelerated method, which was a combination of two groups, denoted by $\Gamma = \{\Gamma_1, \Gamma_2\}$. Every group contained noisy versions of seven images with different density salt-and-pepper noise. The original noise-free images are shown in Figure 4. They include miscellaneous images associated with Γ_1 and texture images associated with Γ_2.

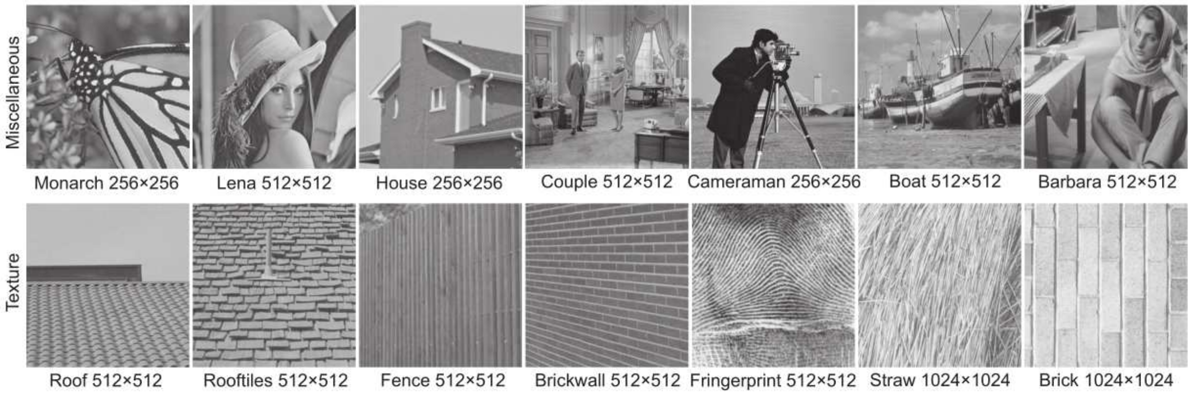


Figure 4. The noise-free images associated with the test set $\Gamma = \{\Gamma_1, \Gamma_2\}$

The accelerated method was applied to the test set Γ. All the parameter values used in the experiments are those recommended in Section 4.2. The PSNR and SSIM results are reported in Table 4 and Table 5, respectively. Moreover, the visual results and zoom-in for two images, House and Roof-tiles, are shown in Figure 5 and Figure 6, respectively.

4.4. **Comparisons.** To augment the performance evaluations, the accelerated method was compared with the MED filter, SMF and SGM. The MED filter uses the sliding window of size 3 × 3. SMF employs the adapted sliding window and selects 70 as judging threshold and output threshold. The source codes of SGM were taken from the original authors, and the parameters used in the experiments were those recommended by the

Table 4. PSNR/SSIM results for miscellaneous images

π	**The miscellaneous images in the group Γ_1**						
	Monarch	Lena	House	Couple	Cameraman	Boat	Barbara
5%	40.72/1.00	46.25/1.00	45.25/0.99	42.06/0.99	38.81/0.99	42.11/0.99	37.79/0.99
10%	37.72/0.99	43.01/0.99	42.22/0.99	38.83/0.99	35.82/0.99	38.97/0.98	34.49/0.98
20%	34.15/0.98	39.66/0.98	38.66/0.98	35.56/0.97	32.25/0.97	35.71/0.97	31.23/0.96
30%	31.81/0.97	37.47/0.97	36.56/0.96	33.45/0.95	30.16/0.96	33.59/0.95	29.32/0.94
40%	29.85/0.96	35.68/0.96	34.71/0.95	31.76/0.93	28.38/0.93	31.97/0.92	27.88/0.91
50%	28.22/0.94	34.06/0.94	33.16/0.93	30.24/0.90	26.94/0.91	30.51/0.90	26.70/0.88
60%	26.50/0.91	32.51/0.92	31.55/0.91	28.85/0.86	25.74/0.88	29.10/0.86	25.69/0.84
70%	24.78/0.87	30.92/0.89	29.89/0.88	27.50/0.81	24.37/0.84	27.73/0.82	24.76/0.80
80%	22.95/0.81	29.04/0.85	27.99/0.83	25.96/0.75	22.99/0.78	26.11/0.76	23.79/0.74
90%	20.37/0.70	20.37/0.70	25.58/0.77	24.02/0.64	21.22/0.70	24.13/0.67	22.64/0.65

TABLE 5. PSNR/SSIM results for texture images

π	**The texture images in the group Γ_2**						
	Roof	Rooftiles	Fence	Brickwall	Fingerprint	Straw	Brick
5%	41.91/0.99	42.34/0.99	45.26/0.99	42.84/0.99	42.21/1.00	43.62/1.00	42.75/0.99
10%	38.69/0.98	39.01/0.99	42.06/0.98	39.67/0.98	38.86/0.99	40.42/0.99	39.64/0.98
20%	35.38/0.96	35.60/0.97	38.92/0.96	36.41/0.96	35.21/0.99	37.07/0.98	36.37/0.96
30%	33.21/0.93	33.31/0.96	36.95/0.94	34.27/0.94	32.67/0.98	34.90/0.97	34.33/0.94
40%	31.49/0.9	31.46/0.94	35.46/0.92	32.58/0.91	30.54/0.96	33.11/0.95	32.73/0.91
50%	30.01/0.87	29.78/0.91	34.18/0.90	31.17/0.88	28.56/0.94	31.47/0.93	31.36/0.87
60%	28.53/0.83	28.03/0.88	32.94/0.86	29.77/0.84	26.51/0.91	29.77/0.90	30.04/0.83
70%	27.04/0.78	26.20/0.83	31.62/0.82	28.27/0.79	24.40/0.86	27.96/0.85	28.73/0.77
80%	25.38/0.71	23.92/0.76	30.04/0.76	26.56/0.72	21.92/0.76	25.70/0.78	27.28/0.70
90%	23.36/0.61	20.99/0.63	27.85/0.66	24.33/0.60	19.05/0.57	22.80/0.63	25.42/0.58

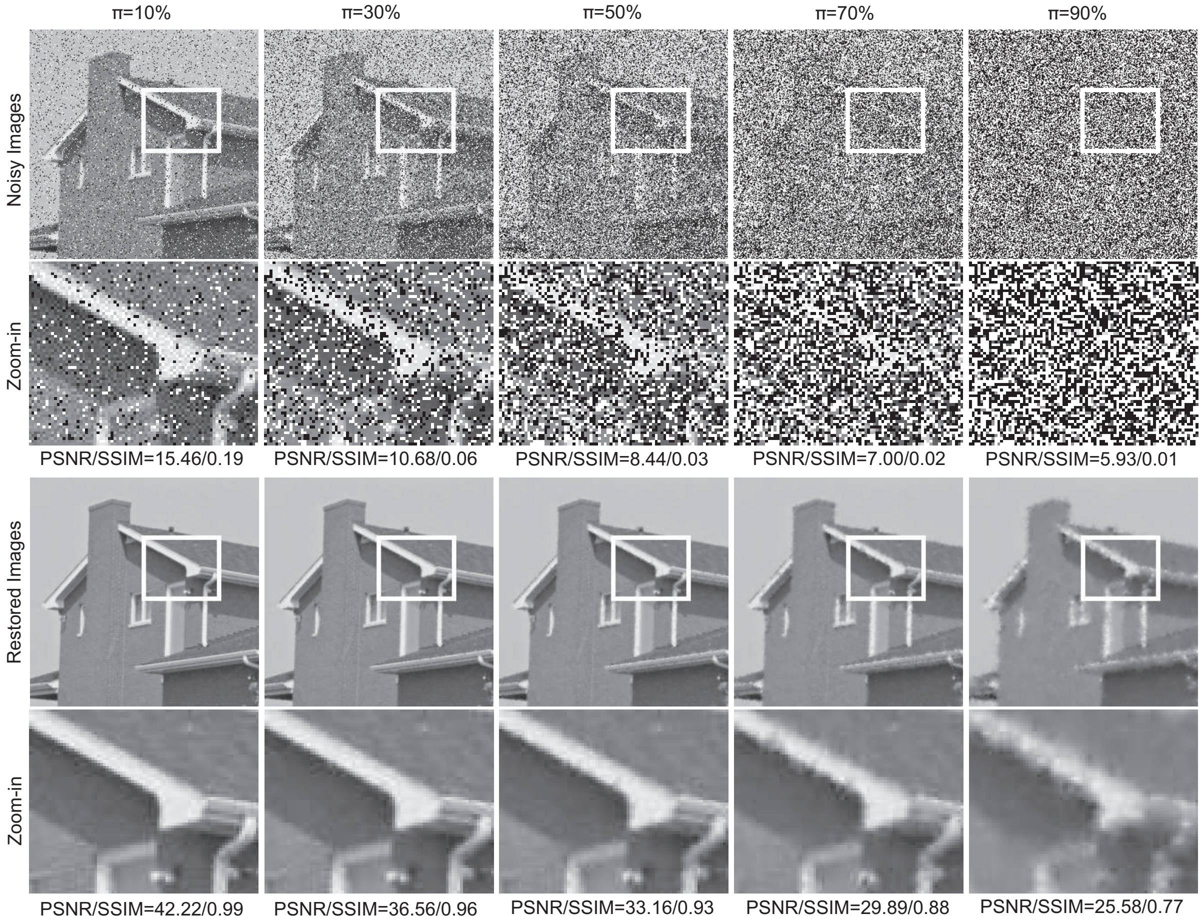


FIGURE 5. Visual results for the House image on five different noise levels

authors. The three methods were also applied to the same test set Γ. Along with the accelerated method and the noisy method, the PSNR/SSIM results from the five methods on the Monarch image are reported in Table 6. The noisy method means the metrics in (19) and (20) use noisy images u_0 rather than restored images $\hat{u}$. Moreover, visual results from the five methods on the Monarch image are shown in Figure 7, and the information about these images is labeled sideways.

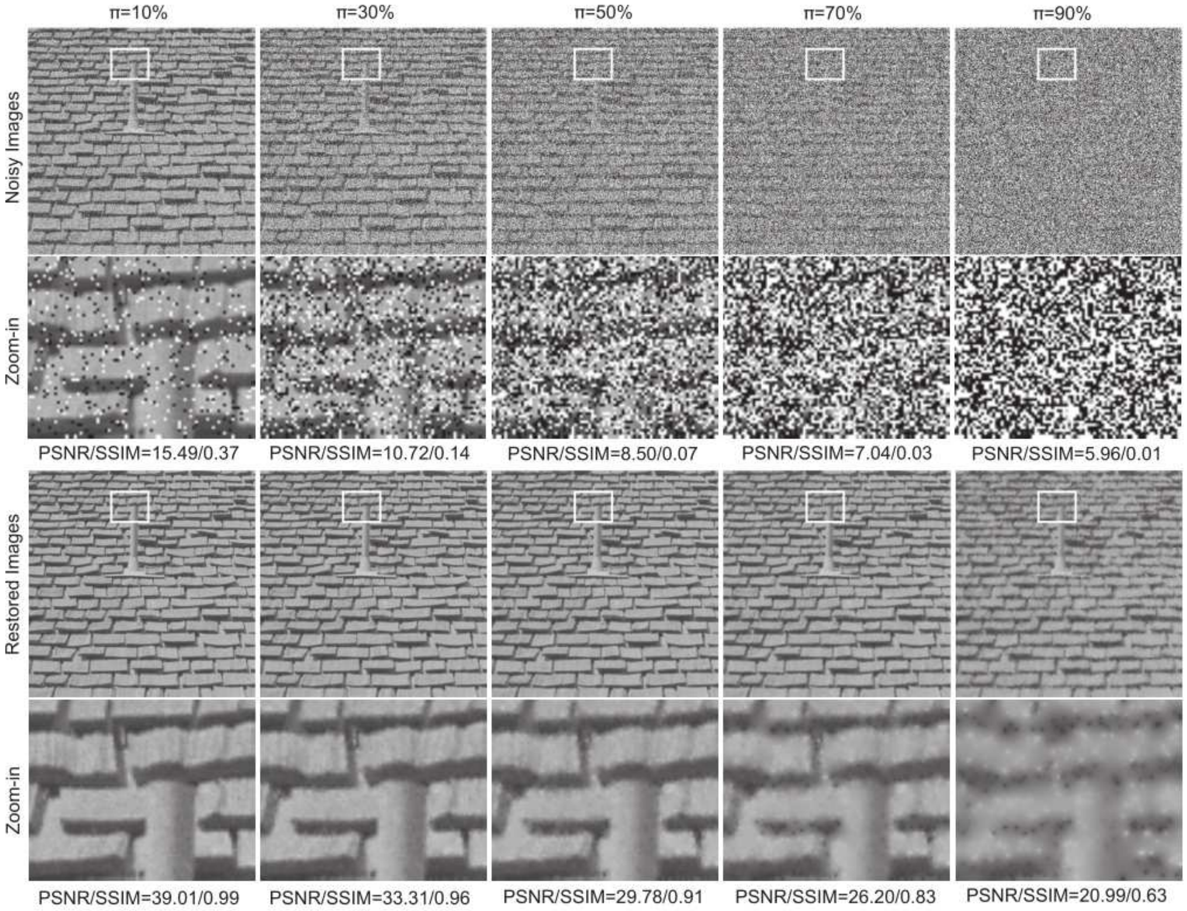


FIGURE 6. Visual results for image Roof-tiles on five different noise levels

TABLE 6. PSNR/SSIM results from different methods for the Monarch image

π	**Different methods**				
	noisy	MED	SMF	SGM	ours
5%	18.41/0.48	29.79/0.95	19.96/0.82	40.45/1.00	40.72/1.00
10%	15.32/0.31	28.39/0.94	19.64/0.77	37.16/0.99	37.72/0.99
20%	12.26/0.19	25.20/0.89	19.01/0.71	34.11/0.98	34.15/0.98
30%	10.51/0.13	21.51/0.76	18.42/0.66	31.07/0.97	31.81/0.97
40%	9.26/0.09	17.90/0.54	17.85/0.62	29.78/0.96	29.85/0.96
50%	8.27/0.07	14.76/0.35	17.19/0.58	28.14/0.94	28.22/0.94
60%	7.50/0.05	12.03/0.20	16.49/0.53	26.31/0.91	26.50/0.91
70%	6.82/0.03	9.76/0.11	15.52/0.47	24.28/0.86	24.78/0.87
80%	6.23/0.02	7.97/0.06	13.90/0.38	21.64/0.78	22.95/0.81
90%	5.73/0.01	6.48/0.03	9.93/0.13	17.63/0.58	20.37/0.70

In addition, the mean PSNR and mean SSIM results for fixed noise are calculated for noisy method, the MED filter, SMF, SGM and the accelerated method, respectively. The calculation is as follows

$$\overline{PSNR}_{\pi} = \frac{1}{|\Gamma_{\pi}|} \sum_{k \in \Gamma_{\pi}} PSNR(k|\pi) \tag{21}$$

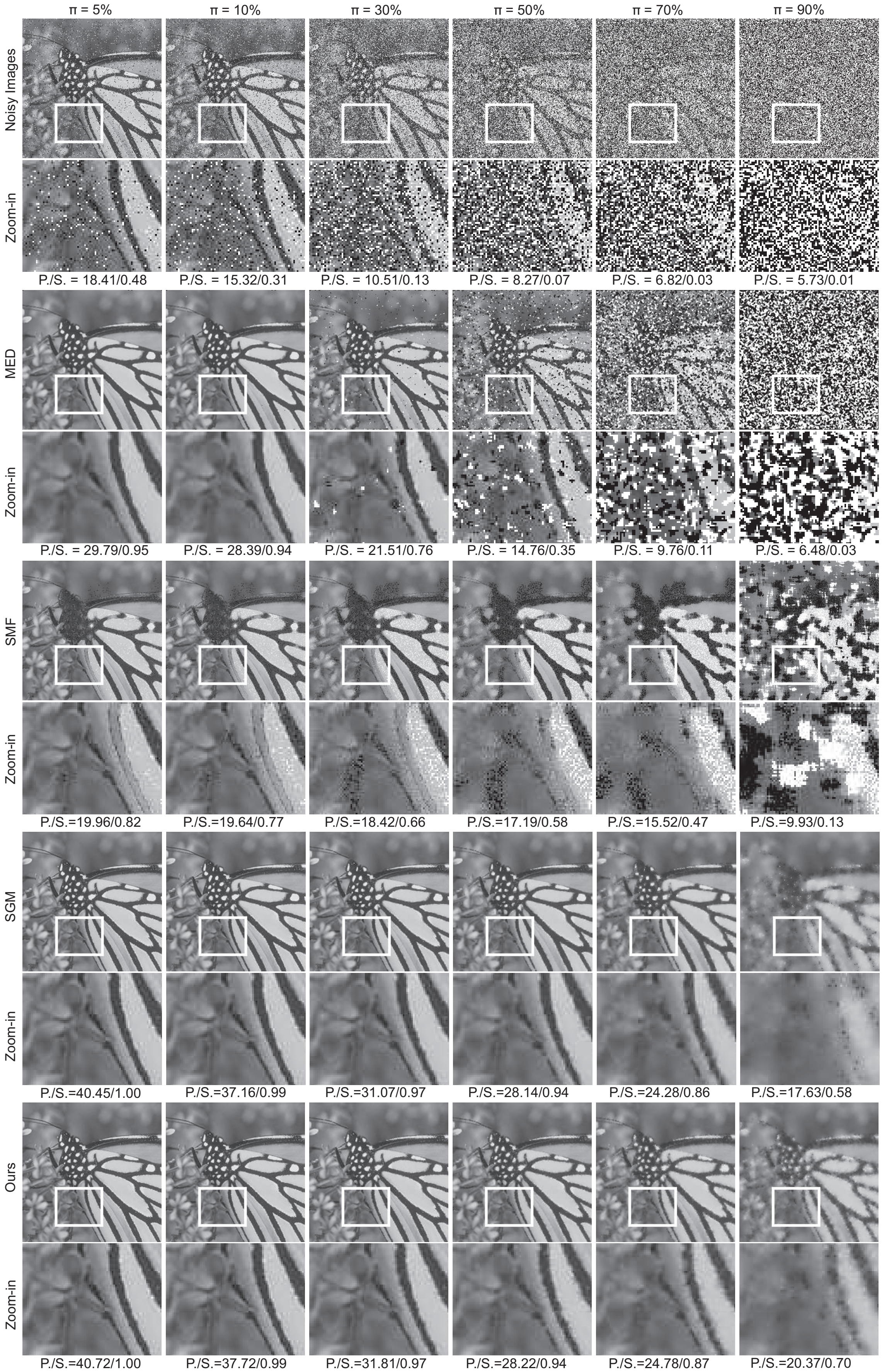


Figure 7. Visual results from five methods on image Monarch

and

$$\overline{SSIM}_{\pi} = \frac{1}{|\Gamma_{\pi}|} \sum_{k \in \Gamma_{\pi}} SSIM(k|\pi), \tag{22}$$

where Γ_{π} denotes all the noisy images with the same density noise π in Γ, and $PSNR_{\pi}(k)$ and $SSIM_{\pi}(k)$ denote the PSNR and SSIM values of the k-th image with the density noise π, respectively. For example, if the method is the accelerated, $\overline{PSNR}_{0.05}$ denotes the mean PSNR for the seven images with the density noise 0.05, corresponding to the accelerated method. The mean PSNR and SSIM values from different methods and on different density noise are plotted in Figure 8.

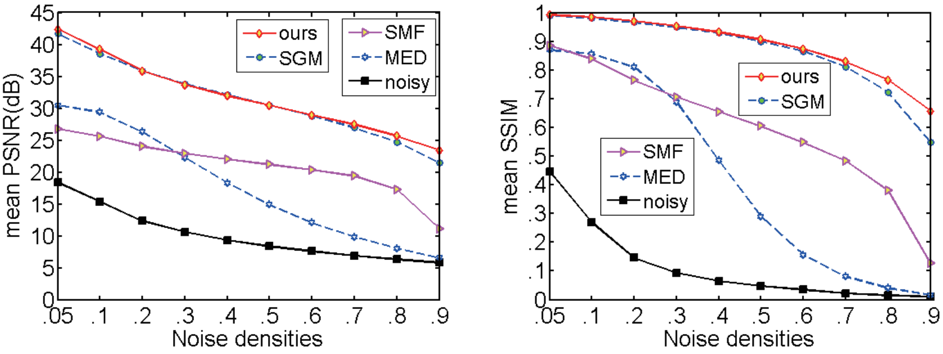


FIGURE 8. The mean PSNR and SSIM results from different methods and on different density noise

The following observations and conclusions from the quantitative measurements and visual results can be drawn. Firstly, the accelerated method achieved the best results in every case tested, as seen in Table 6 or Figure 8. It achieved 0.48dB improvement over the SGM method on average, and significantly outperformed the SMF and the MED filter by 10.86dB and 14.11dB, respectively, in the PSNR results and on the test set Γ. The accelerated method achieved 0.02 improvement over the SGM method on average, and significantly outperformed the SMF and the MED filter by 0.29 and 0.46, respectively, in the SSIM results and also on the test set Γ. Secondly, the accelerated method has a strong capability to preserve details. The accelerated method reconstructed more image details from noisy images than the MED filter, SMF and SGM. The MED filter failed to reconstruct images when the noise density was larger than 50; the SMF introduced too much smoothing and resulted in blurred visual features, as seen in Figure 7. The SGM blurred more visual features than the accelerated method when the noise density was 90%, as seen in Figure 7. Thirdly, the proposed method is more robust to different noise strengths than the MED filter, SMF and SGM. In summary, our method shows strong capability to reduce noise in terms of the PSNR/SSIM results and visual perception quality in restored images.

5. **Conclusions and Future Directions.** In this paper, a modified PM diffusion method is proposed to reduce salt-and-pepper noise for gray scale images. The proposed method belongs to switching filtering, containing noise detection module and noise removal module. The former identifies corrupted pixels, and the latter applies the modified PM diffusion method to these corrupted pixels. In contrast to original PM diffusion method, the modified PM diffusion only treats corrupted pixels rather than all pixels in an image domain, and conducts along eight-neighbors directions rather than along four-neighbors directions. Since the first modification, the modified PM method is suitable for images

not only with low noise but also with high noise; since the second modification, the modified PM method can reconstruct more details from noisy images. To reduce the number of iterations during processing, the original initialization image (the noisy image) is replaced by an evolved intermediate image. In our experiments, different images were tested, containing 70 miscellaneous and 70 texture images with different density noise. Experimental results show that the method has a very strong capability to suppress noise in terms of both quantitative measurement and visual perception quality of restored images.

Although the modified PM diffusion method can achieve good denoising results for gray-scale images with salt-and-pepper noise, it does not work well on random valued impulses. A main reason is that the devised detector is suitable for salt-and-pepper noise; however, it must be improved for random valued impulses. In addition, the modified PM diffusion method is applied in two-dimensions; how can we apply this technique in three-dimensions, and serve for multi-channels color images? All these issues are the future directions to our striving.

Acknowledgment. This work was supported by Chongqing Research Program of Application Foundation and Advanced Technology (under grant No. cstc2016jcyjA0571).

REFERENCES

[1] H. L. Yong and S. A. Kassam, Generalized median filtering and related nonlinear filtering techniques, *IEEE Trans. Acoustics Speech & Signal Processing*, vol.ASSP-33, no.3, pp.672-683, 1985.

[2] I. Pitas and A. N. Venetsanopoulos, Order statistics in digital image processing, *Proc. of the IEEE*, vol.80, no.12, pp.1893-1921, 1992.

[3] D. R. K. Brownrigg, The weighted median filter, *Communications of the ACM*, vol.27, no.8, pp.807-818, 1984.

[4] S. J. Ko and H. L. Yong, Center weighted median filters and their applications to image enhancement, *IEEE Trans. Circuits & Systems*, vol.38, no.9, pp.984-993, 1991.

[5] H. Hwang and R. A. Haddad, Adaptive median filters: New algorithms and results, *IEEE Trans. Image Processing: A Publication of the IEEE Signal Processing Society*, vol.4, no.4, pp.499-502, 1995.

[6] M. Muneyasu, N. Nishi and T. Hinamoto, A new adaptive center weighted median filter using counter propagation networks, *Journal of the Franklin Institute*, vol.337, no.5, pp.631-639, 2000.

[7] T. Sun and Y. Neuvo, Detail-preserving median based filters in image processing, *Pattern Recognition Letters*, vol.15, no.4, pp.341-347, 1994.

[8] Y. Y. Zhou, Z. F. Ye and J. J. Huang, Improved decision-based detail-preserving variational method for removal of random-valued impulse noise, *IET Image Processing*, vol.6, no.7, pp.976-985, 2012.

[9] J. Matsuoka, T. Koga, N. Suetake and E. Uchino, Switching non-local median filter, *Optical Review*, vol.22, no.3, pp.448-458, 2015.

[10] R. Garnett, T. Huegerich, C. Chui and W. He, A universal noise removal algorithm with an impulse detector, *IEEE Trans. Image Processing*, vol.14, no.11, pp.1747-1754, 2005.

[11] H. Dawood, H. Dawood and P. Guo, Removal of random-valued impulse noise by local statistics, *Multimedia Tools and Applications*, vol.74, no.24, pp.11485-11498, 2015.

[12] M. Habib, A. Hussain, S. Rasheed and M. Ali, Adaptive fuzzy inference system based directional median filter for impulse noise removal, *AEU – International Journal of Electronics and Communications*, vol.70, no.5, pp.689-697, 2016.

[13] C. T. Lu and T. C. Chou, Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter, *Pattern Recognition Letters*, vol.33, no.10, pp.1287-1295, 2012.

[14] C. Budak, M. Turk and A. Toprak, Removal of impulse noise in digital images with naive Bayes classifier method, *Turkish Journal of Electrical Engineering and Computer Sciences*, vol.24, no.4, pp.2717-2729, 2016.

[15] G. Kaliraj and S. Baskar, An efficient approach for the removal of impulse noise from the corrupted image using neural network based impulse detector, *Image and Vision Computing*, vol.28, no.3, pp.458-466, 2010.

[16] Y. Li, J. Sun and H. Luo, A neuro-fuzzy network based impulse noise filtering for gray scale images, *Neurocomputing*, vol.127, pp.190-199, 2014.

[17] A. Roy and R. H. Laskar, Multiclass SVM based adaptive filter for removal of high density impulse noise from color images, *Applied Soft Computing*, vol.46, pp.816-826, 2016.

[18] B. A. Weyori, K. O. Boateng, P. K. Yeboah and P. N. Amponsah, Design and implementation of the block matching hybrid median filter for noise removal in color images, *International Journal of Innovative Computing, Information and Control*, vol.12, no.6, pp.1929-1941, 2016.

[19] F. Chen, G. Ma, L. Lin and Q. Qin, Impulsive noise removal via sparse representation, *Journal of Electronic Imaging*, vol.22, no.22, pp.451-459, 2013.

[20] J. K. Liu and S. J. Li, Spectral gradient method for impulse noise removal, *Optimization Letters*, vol.9, no.7, pp.1341-1351, 2015.

[21] F. Dong, Y. Chen, D.-X. Kong and B. Yang, Salt and pepper noise removal based on an approximation of norm, *Computers & Mathematics with Applications*, vol.70, no.5, pp.789-804, 2015.

[22] A. P. Witkin, Scale-space filtering, *Proc. of the 8th International Joint Conference on Artificial Intelligence*, vol.2, pp.1019-1022, 1983.

[23] P. Perona and J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol.12, no.7, pp.629-639, 1990.

[24] F. Catte, P. L. Lions, J. M. Morel and T. Coll, Image selective smoothing and edge-detection by nonlinear diffusion, *SIAM Journal on Numerical Analysis*, vol.29, no.1, pp.182-193, 1992.

[25] Y. L. You and M. Kaveh, Fourth-order partial differential equations for noise removal, *IEEE Trans. Image Processing*, vol.9, no.10, pp.1723-1730, 2000.

[26] J. Yu, J. Tan and Y. Wang, Ultrasound speckle reduction by a SUSAN-controlled anisotropic diffusion method, *Pattern Recognition*, vol.43, no.9, pp.3083-3092, 2010.

[27] H. C. Li, P. Z. Fan and M. K. Khan, Context-adaptive anisotropic diffusion for image denoising, *Electronics Letters*, vol.48, no.14, pp.827-829, 2012.

[28] X. Zhang and S. Zhang, Diffusion scheme using mean filter and wavelet coefficient magnitude for image denoising, *AEU International Journal of Electronics and Communications*, vol.70, no.7, pp.944-952, 2016.

[29] D. Gupta, R. S. Anand and B. Tyagi, Despeckling of ultrasound medical images using nonlinear adaptive anisotropic diffusion in nonsubsampled shearlet domain, *Biomedical Signal Processing & Control*, vol.14, no.14, pp.55-65, 2014.