Hindawi
Journal of Sensors
Volume 2017, Article ID 7635641, 18 pages
https://doi.org/10.1155/2017/7635641

Research Article

Decision-Based Marginal Total Variation Diffusion for Impulsive Noise Removal in Color Images

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

[1] *School of Information & Software Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2] *College of Computer Engineering, Yangtze Normal University, Chongqing 408000, China*
[3] *School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

Correspondence should be addressed to Hongyao Deng; hydeng_2004@163.com

Received 20 January 2017; Revised 19 March 2017; Accepted 8 May 2017; Published 19 June 2017

Academic Editor: Bruno Andò

Impulsive noise removal for color images usually employs vector median filter, switching median filter, the total variation L_1 method, 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 density noise. A marginal method to reduce impulsive noise is proposed in this paper that overcomes this limitation that is based on the following facts: (i) each channel in a color image is contaminated independently, and contaminative components are independent and identically distributed; (ii) in a natural image the gradients of different components of a pixel are similar to one another. This method divides components into different categories based on different noise characteristics. If an image is corrupted by salt-and-pepper noise, the components are divided into the corrupted and the noise-free components; if the image is corrupted by random-valued impulses, the components are divided into the corrupted, noise-free, and the possibly corrupted components. Components falling into different categories are processed differently. If a component is corrupted, modified total variation diffusion is applied; if it is possibly corrupted, scaled total variation diffusion is applied; otherwise, the component is left unchanged. Simulation results demonstrate its effectiveness.

1. Introduction

Noise reduction is a fundamental issue in image processing. In order to reduce noise, a great number of works have been presented over the past several decades. The overwhelming majority of these works cope with the gray-scale images. With the increase in use of color images, however, more and more works to reduce noise for color images are rapidly growing.

Indeed, color images often are corrupted by various types of noise. In this paper, impulsive noise is only considered, containing salt-and-pepper noise and random-valued impulses where they present themselves as occurring isolated chromatic points. To yield better images from their noisy versions, a series of various methods have been proposed.

Vector median filter (VMF) [1] and variants, based on reduced ordering, are popular methods. VMF calculates the cumulative sums of the Euclidean distance from every pixel to other pixels in a filtering window and outputs the pixel corresponding to the minimization sum of distances. The output pixel is called vector median, since each pixel is a vector containing multiple components. To achieve better performance in the variants, a few dissimilarity measures are used other than the Euclidean distance. For example, Basic Vector Directional Filter (BVDF) uses the angular distance [2]; Directional Distance Filter (DDF) employs the combination of the magnitude and directional processing [3]. In addition, α-Trimmed Vector Median Filter (T_αVMF) only considers alpha pixels to calculate the dissimilarity [4], rather than all pixels within a filtering window. These methods process the noisy and noise-free pixels uniformly. Therefore they introduce extensive smoothing and blur visual features extensively and are only suited to low density noise.

To overcome this limitation, various kinds of switching vector filters have been proposed in the rich literature, which aim to only replace the corrupted pixels. This type of filters usually consists of noise detection and noise removal. The former identifies the corrupted pixels, and the latter applies some algorithm to those corrupted pixels. The noise detection is crucial. If a detector fails to identify corrupted pixels, then they will be left unchanged, resulting in poor filtered images. If the detector classifies a corrupted pixel correctly but also identifies noise-free pixels as corrupted, the tiny details may be lost. Therefore, accurate and complete identification determines the quality of filtered images.

Currently, there exist many techniques to identify noisy pixels. Peris-Fajarnes et al. [5] use the rank-ordered differences statistic method. Smolka [6] uses a soft switching technique that assigns trimmed cumulated distances to the central pixel, where the trimmed sum of distances serves as an indicator manifesting the corrupted extent of the current pixel. Morillas et al. [7] utilize the local self-adaptive fuzzy filter. Ananthi and Balasubramaniam [8] use the technique of the interval-valued fuzzy set. Jin et al. [9] employ the quaternion theory. Malinski and Smolka [10] use adaptive switching technique. Particularly, after the concept of a peer group is introduced, many schemes based on it have been presented; for example, Morillas et al. [11] use the fuzzy peer groups, and Malinski and Smolka [12] utilize the fast averaging peer group.

Apart from the filters mentioned above, there exist a family of variational and partial differential equations. This type of methods usually adopts a variational energy minimization model to obtain the solution, such as the well-known Rudin-Osher-Fatemi (ROF) model [13], the hybrid model [14], the modified total variation diffusion method [15], and the total variation L_1 model [16]. The main advantage for them is to use the gradient of images to represent the model, and thus they have the capability of edge-preserving. Unfortunately, this type of methods has the drawback similar to the VMF; that is, they only work well on the low density noise because they do not distinguish between the corrupted and the uncorrupted pixels.

Inspired by switching filters and variational methods, in this paper a decision-based marginal diffusion method is proposed to reduce impulsive noise for color images. The proposed method, in contrast to vectorial methods emphasizing the correlation between the channels, independently treats each channel in a color image and implements diffusion operations on every corrupted component rather than every vector. In addition, the proposed method divides components into different categories based on different noise characteristics. If an image is corrupted by salt-and-pepper noise, the components are divided into the corrupted and the noise-free components; if the image is corrupted by random-valued impulses, the components are divided into the corrupted, noise-free, and the possibly corrupted components. Components falling into different categories are processed differently. If a component is corrupted, a modified total variation diffusion is applied; if the component is possibly corrupted, a scaled total variation diffusion is applied; otherwise, the component is left unchanged. Experimental results show that the proposed method is robust to different noise strengths and suitable for different images, with strong noise removal capability as shown by PSNR/MSSIM/FSIM results as well as the visual quality of restored images.

The rest of this paper is organized as follows. In Section 2, the details of noise removal are introduced. Section 3 reports experimental results and comparisons, and conclusions for this paper are drawn in Section 4.

2. The Proposed Denoising Method

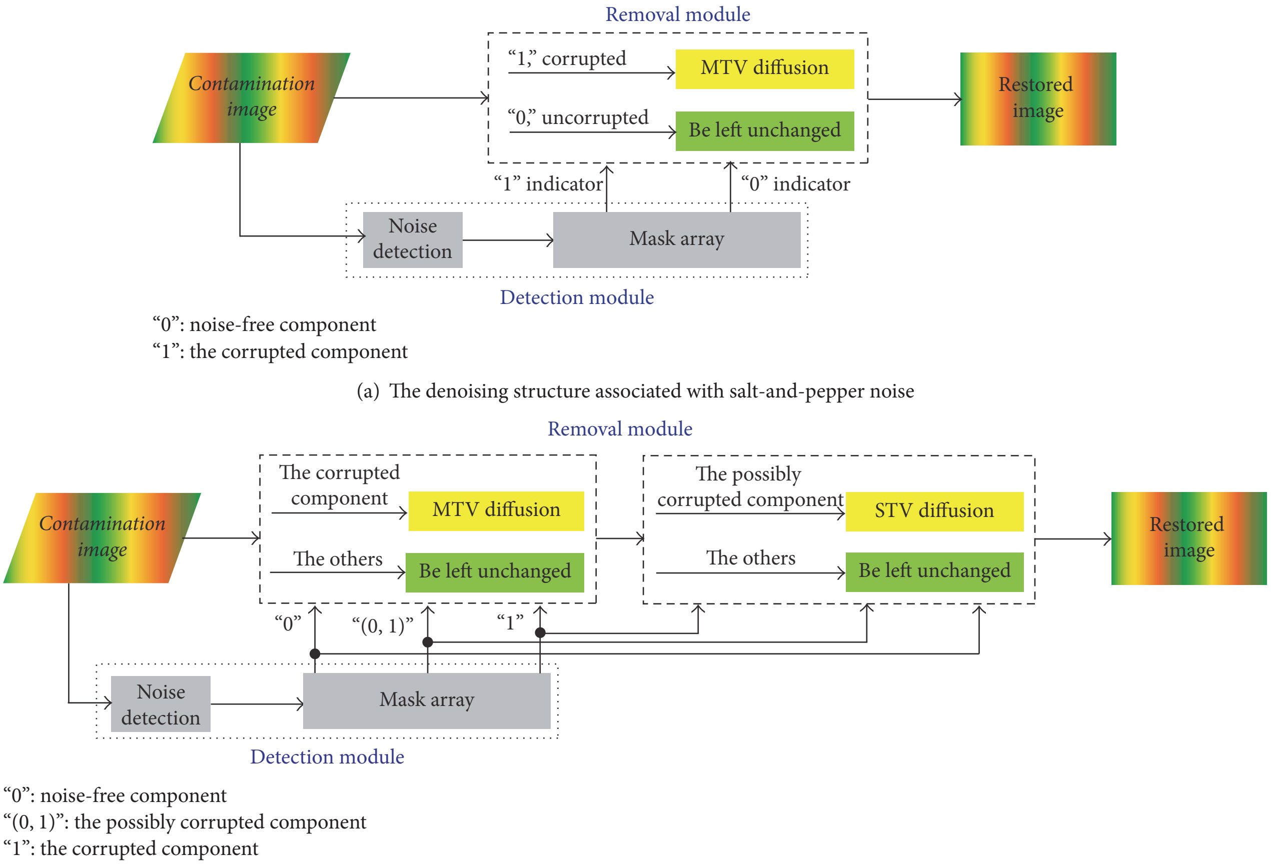
In this section, the details of reducing salt-and-pepper noise and random-valued impulses are described, containing denoising structures, noise models, noise detection, and noise removal.

2.1. Noise Removal Structure. Two denoising structures are devised based on two noise characteristics, respectively. As shown in Figure 1, inset (a) is associated with salt-and-pepper noise and inset (b) is associated with random-valued impulses. Both structures consist of noise detection module and noise removal module, corresponding to detection stage and removal stage. For salt-and-pepper noise, a detector divides components into the corrupted and the noise-free components; for random-valued impulses, another detector divides components into the corrupted, noise-free, and the possibly corrupted components. Based on the two detection results, two mask arrays are built where each entry is a label indicating which category the corresponding component belongs to. Components falling into different categories are processed differently in separate channels. If a component is corrupted, modified total variation (MTV) diffusion is applied; if the component is possibly corrupted, scaled total variation (STV) diffusion is applied; otherwise, the component is left unchanged. To achieve better results, diffusion operations are iteratively implemented following the corresponding mask array instructions. Especially, for random-valued impulses, a hierarchical scheme is adopted in removal stage; the corrupted components are first processed and then the possibly corrupted ones, as in inset (b).

2.2. Noise Model and Analysis. RGB images to be processed are only considered in this paper, which contain three channels, the red, green, and blue, and a color image is defined as a two-dimensional matrix consisting of a certain number of pixels. Let $\mathbf{u}_{i,j} = (u_{i,j}^1, u_{i,j}^2, u_{i,j}^3)$ denote a pixel at the position (i, j) in the matrix and components $u_{i,j}^q$ for $q = 1, 2, 3$ denote the RGB channel intensity values; a contamination image can be modeled as

$$u_{i,j}^q = \begin{cases} v_{i,j}^q & \text{with probability } \pi \\ o_{i,j}^q & \text{with probability } 1 - \pi \end{cases}, \quad \text{for } q = 1, 2, 3, \quad (1)$$

where $o_{i,j}^q$ and $v_{i,j}^q$ denote the original component and the contaminative component, respectively. In this model, the intensity of component $v_{i,j}^q$ is a random variable. If $v_{i,j}^q$ takes the value 0 or 255 with equal probability, it is salt-and-pepper noise model, denoted by SPM; if $v_{i,j}^q$ takes any value from the range $[0, 255]$, then it is random-valued impulsive noise model, denoted by RDM, assuming 8-bit per channel image representation.



(a) The denoising structure associated with salt-and-pepper noise

(b) The denoising structure associated with random-valued impulses

Figure 1: Two structures of the noise removal. The operations surrounded by the rectangle with dashed lines are repeated.

The probability of a noise-free component processed incorrectly is analyzed in the vectorial and marginal methods. Assuming detectors can correctly judge whether every component is corrupted or not, then the marginal methods can correctly process all components in the removal stage. The vectorial methods, however, may identify noise-free components as noisy to process, because the processing is based on the pixel unit rather than the component unit. If a component is corrupted with probability π, then a pixel is not corrupted with probability $(1-\pi)^3$ since a pixel may have corrupted 1, 2, or all 3 channels. Therefore, in vectorial methods a noise-free component is processed incorrectly with probability

$$p(\pi) = \frac{\left[1-(1-\pi)^3\right]-\pi}{1-(1-\pi)^3} = \frac{2-3\pi+\pi^2}{3-3\pi+\pi^2}. \quad (2)$$

As shown in Figure 2, with the increasing probability π, the probability of a noise-free component processed incorrectly is decreased. In an extreme case, only when all components are corrupted, can every component be correctly processed. Just from this viewpoint, the marginal methods are superior to the vectorial methods.

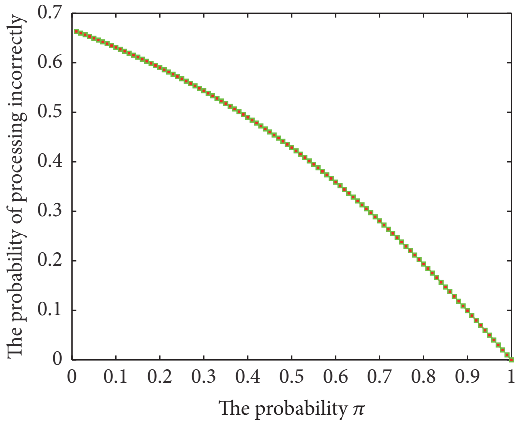


Figure 2: Illustration of the relationship between two probabilities, where a component is corrupted and a noise-free component is processed incorrectly, assuming a vectorial method is applied.

2.3. Noise Detection. Two detectors are used in the SPM and RDM, respectively. Both work in separate channels, based on the features of local neighborhood. In other words, a component is detected within the channel it lies in. Let $u^q_{x,y}$ be a current component centered in the sliding window; its neighborhood is defined as

$$N^q_{x,y}(R) = \{(i,j) : |x-i| \le R,\ |y-j| \le R\}, \qquad \text{for } (x,y) \in \Omega^q,\ q = 1,2,3, \tag{3}$$

where R is the neighborhood radius quantified into the positive integer domain and Ω^q denotes the qth channel domain. Obviously, a neighborhood contains $(2R+1)\times(2R+1)$ samples.

The Detector Used in SPM. This detector takes components with intensity 0 or 255 as the noisy candidates. Let $u^q_{x,y}$ be a candidate, the absolute differences of intensities between it and every sample within the neighborhood are calculated, and then the number of samples, whose absolute difference is larger than a predefined threshold value, is determined, expressed as

$$m^q_{x,y} = \#\left\{u^q_{i,j} : (i,j) \in N^q_{x,y}(R),\ \left|u^q_{x,y} - u^q_{i,j}\right| > T\right\}, \tag{4}$$

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

The number $m_{x,y}$ is used to determine if a current candidate is corrupted. The determining function is

$$f(x,y) = \begin{cases} 1, & \text{if } m^q_{x,y} > \text{thr} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

In this formula, if $f(x,y) = 1$, then candidate $u^q_{x,y}$ is corrupted; otherwise, it is uncorrupted. When all candidates are complete, a binary mask array can be built; each entry is either 1 or 0.

The Detector Used in RDM. This detector takes all components as the noisy candidates, which is devised based on the Rank-Ordered Absolute Differences (ROAD) [19]. It assigns an index to each candidate. Based on these indices, the corresponding candidate is judged as noise-free, corrupted, or possibly corrupted. Let $u^q_{x,y}$ be the current candidate; the ROAD of $u^q_{x,y}$ is defined by

$$\text{ROAD}^q_\alpha(x,y) = \sum_{k=1}^{\alpha} r^q_k(x,y), \quad \text{for } q = 1,2,3, \tag{6}$$

where $2 \le \alpha \le (2R+1)\times(2R+1)$ is a integer, and

$$r^q_k(x,y) = k^{\text{th}} \text{ smallest } \left|u^q_{x,y} - u^q_{i,j}\right|, \qquad \text{for } (i,j) \in N^q_{x,y}(R). \tag{7}$$

Then, the index of $u^q_{x,y}$ is

$$h^q(x,y) = \begin{cases} 0, & \text{ROAD}^q_\alpha(x,y) \le T_1 \\ \dfrac{\text{ROAD}^q_\alpha(x,y) - T_1}{T_2 - T_1}, & T_1 < \text{ROAD}^q_\alpha(x,y) < T_2, \\ 1, & \text{ROAD}^q_\alpha(x,y) \ge T_2, \end{cases} \tag{8}$$

where T_1 and T_2 are predefined threshold parameters. The index $h^q(x,y)$, for $(x,y) \in \Omega^q$, is a determining function, which divides candidates into three categories. If $h^q(x,y) = 0$, then $u^q_{x,y}$ is declared as noisy-free; if $h^q(x,y) = 1$, $u^q_{x,y}$ is declared as the corrupted component; otherwise, it is a possibly corrupted component.

2.4. Noise Removal. In the removal stage, diffusion operations are implemented in separate channels. The main reason is twofold. First, each channel in a color image is contaminated independently, and contaminative components are independent and identically distributed. Second, in a natural image the gradients of different components of a pixel are similar to one another, where Figure 3 illustrates this property.

Moreover, components falling into different categories are processed in different diffusion ways. In SPM, the components are divided into the corrupted and noise-free components. If a component is corrupted, the MTV diffusion is applied; otherwise, the component is left unchanged. Let u^q be the qth channel of the noise-free image, u^q_0 be its noisy version, and $\Theta^q \subset \Omega^q$ be the qth distorted channel domain where $\Omega^q \subset \Omega$ is the qth channel domain and $\Omega = \{\Omega^1, \Omega^2, \Omega^3\}$; then the MTV diffusion is expressed as

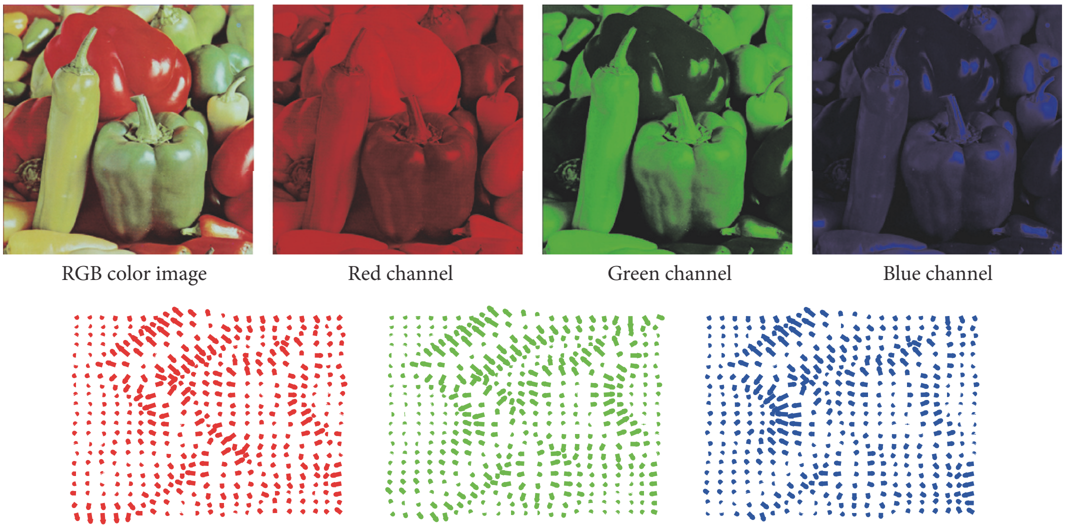
$$\text{MTV}(u^q) = \iint_{\Theta^q} |\nabla u^q|\, dx\, dy. \tag{9}$$

Applying Euler-Lagrange to (9), the corresponding descent flow is

$$\begin{aligned} \frac{\partial}{\partial t} u^q(x,y,t) &= \operatorname{div}\left(\frac{1}{|\nabla u^q|_\beta} \cdot \nabla u^q\right), \qquad \text{for } (x,y) \in \Theta^q \\ u^q(x,y,0) &= u^q_0, \end{aligned} \tag{10}$$

where t and div are the time step and the divergence operator and $|\nabla u^q|_\beta = \sqrt{|\nabla u^q|^2 + \beta}$, where β is a positive lifting parameter avoiding the $|\nabla u^q|$ vanishing.

In RDM, the components are divided into the corrupted, noise-free, and the possibly corrupted components, and a hierarchical scheme is adopted that first processes the corrupted and then the possibly corrupted ones. As shown in inset (b) in Figure 2, in the first stage, if a component is corrupted, the MTV diffusion is applied; otherwise, the component is left unchanged. In the second stage, if a component is possibly corrupted, the STV diffusion is applied;



RGB color image

Red channel

Green channel

Blue channel

Gradient of red component

Gradient of green component

Gradient of blue component

Figure 3: Illustration of RGB three channels and gradients of component in separate channels.

otherwise, the component is left unchanged. Let $\Theta^{q[v]}$ denote the corrupted domain of the qth channel; the descent flow of MTV in RDM is as follows:

$$\frac{\partial}{\partial t} u^q(x,y,t) = \operatorname{div}\left(\frac{1}{|\nabla u^q|_\beta}\cdot \nabla u^q\right), \quad \text{for } (x,y)\in\Theta^{q[v]} \qquad (11)$$
$$u^q(x,y,0) = u_0^q.$$

Let $\Theta^{q[p]}$ denote the possibly corrupted domain of the qth channel where $\Theta^q = \{\Theta^{q[v]}, \Theta^{q[p]}\}$ and u_v^q denotes the qth estimated channel by the MTV diffusion; then the descent of STV is

$$\frac{\partial}{\partial t} u^q(x,y,t) = h^q(x,y)\operatorname{div}\left(\frac{1}{|\nabla u^q|_\beta}\cdot \nabla u^q\right), \quad \text{for } (x,y)\in\Theta^{q[p]} \qquad (12)$$
$$u^q(x,y,0) = u_v^q,$$

where $h^q(x,y)$ is the scaled parameter, and the parameter value is obtained according to (8):

$$h^q(x,y) = \frac{\mathrm{ROAD}_\alpha^q(x,y) - T_1}{T_2 - T_1}, \qquad T_1 < \mathrm{ROAD}_\alpha^q(x,y) < T_2. \qquad (13)$$

Discretization of the divergence $\operatorname{div}(\nabla u^q/|\nabla u^q|_\beta)$ must be discussed. As shown in Figure 4, the techniques of central finite difference and half-pixel resolution discretization are used. Let $u_{i,j}^q$ be a current component; the divergence at the current $u_{i,j}^q$ is expressed as

$$\begin{aligned}\operatorname{div}\left(\frac{\nabla u^q}{|\nabla u^q|_\beta}\right) &= \frac{\partial}{\partial x}\left(\frac{{u^q}_x}{|\nabla u^q|_\beta}\right) + \frac{\partial}{\partial y}\left(\frac{{u^q}_y}{|\nabla u^q|_\beta}\right)\\ &\simeq \left[\left(\frac{{u^q}_x}{|\nabla u^q|_\beta}\right)_{i+1/2,j} - \left(\frac{{u^q}_x}{|\nabla u^q|_\beta}\right)_{i-1/2,j}\right]\\ &\quad + \left[\left(\frac{{u^q}_y}{|\nabla u^q|_\beta}\right)_{i,j+1/2} - \left(\frac{{u^q}_y}{|\nabla u^q|_\beta}\right)_{i,j-1/2}\right],\end{aligned} \qquad (14)$$

where ${u^q}_x$ and ${u^q}_y$ are the first-order derivatives in the x and y directions in the qth channel, respectively.

At the half-pixel $(i, j+1/2)$ in the qth channel, the following three formulas hold:

$$\begin{aligned}(\nabla u^q)_{i,j+1/2} &= \left(({u^q}_x)_{i,j+1/2}, ({u^q}_y)_{i,j+1/2}\right)\\ ({u^q}_y)_{i,j+1/2} &= u^q_{i,j+1} - {u^q}_{i,j}\\ ({u^q}_x)_{i,j+1/2} &= \frac{\left(u^q_{i+1,j+1/2} - u^q_{i-1,j+1/2}\right)}{2}\\ &= \frac{\left(u^q_{i+1,j+1} + u^q_{i+1,j} - u^q_{i-1,j} - u^q_{i-1,j+1}\right)}{4}.\end{aligned} \qquad (15)$$

In addition, $|\nabla u^q|_\beta = \sqrt{({u^q}_x)^2 + ({u^q}_y)^2 + \beta}$; then

$$\begin{aligned}\left(\frac{{u^q}_y}{|\nabla u^q|_\beta}\right)_{i,j+1/2} &= \frac{({u^q}_y)_{i,j+1/2}}{\sqrt{({u^q}_x)^2_{i,j+1/2} + ({u^q}_y)^2_{i,j+1/2} + \beta}}\\ &= C^q_{1,(i,j)}\left(u^q_{i,j+1} - u^q_{i,j}\right),\end{aligned} \qquad (16)$$

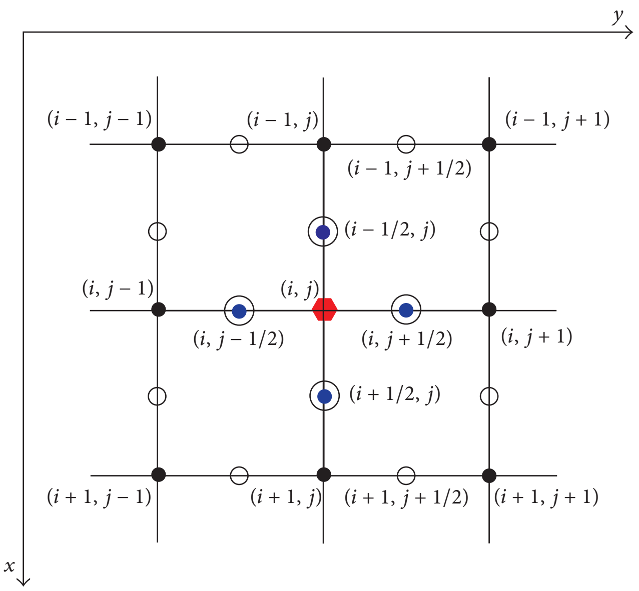


Figure 4: The representation of discretization at half-pixel resolution in a channel.

where

$$C^q_{1,(i,j)} = \frac{1}{\sqrt{\left(u^q_{i+1,j+1} + u^q_{i+1,j} - u^q_{i-1,j} - u^q_{i-1,j+1}\right)^2/16 + \left(u^q_{i,j+1} - u^q_{i,j}\right)^2 + \beta}}. \tag{17}$$

Similarly, at the other three half-pixels,

$$\begin{gathered}\left(\frac{{u^q}_y}{|\nabla u^q|_\beta}\right)_{i,j-1/2} = C^q_{2,(i,j)}\left(u^q_{i,j} - u^q_{i-1,j}\right), \\ \left(\frac{{u^q}_x}{|\nabla u^q|_\beta}\right)_{i+1/2,j} = C^q_{3,(i,j)}\left(u^q_{i+1,j} - u^q_{i,j}\right), \\ \left(\frac{{u^q}_x}{|\nabla u^q|_\beta}\right)_{i-1/2,j} = C^q_{4,(i,j)}\left(u^q_{i,j} - u^q_{i-1,j}\right),\end{gathered} \tag{18}$$

where

$$\begin{gathered}C^q_{2,(i,j)} = \frac{1}{\sqrt{\left(u^q_{i+1,j-1} + u^q_{i+1,j} - u^q_{i-1,j-1} - u^q_{i-1,j}\right)^2/16 + \left(u^q_{i,j-1} - u^q_{i,j}\right)^2 + \beta}}, \\ C^q_{3,(i,j)} = \frac{1}{\sqrt{\left(u^q_{i+1,j+1} + u^q_{i,j+1} - u^q_{i+1,j-1} - u^q_{i,j-1}\right)^2/16 + \left(u^q_{i+1,j} - u^q_{i,j}\right)^2 + \beta}}, \\ C^q_{4,(i,j)} = \frac{1}{\sqrt{\left(u^q_{i,j+1} + u^q_{i-1,j+1} - u^q_{i,j-1} - u^q_{i-1,j-1}\right)^2/16 + \left(u^q_{i-1,j} - u^q_{i,j}\right)^2 + \beta}}.\end{gathered} \tag{19}$$

Substituting (16) and (18) to (14), the divergence $\mathrm{div}(\nabla u^q / |\nabla u^q|_\beta)$ at the current $u^q_{i,j}$ can be discretized as

$$\mathrm{div}\left(\frac{\nabla u^q}{|\nabla u^q|_\beta}\right)_{i,j} = \sum_{k=1}^{4}\left[C^q_{k,(i,j)}\left(P^q_{k,(i,j)} - u^q_{i,j}\right)\right], \tag{20}$$

where $P^q_{k,(i,j)}$, for $k = 1, 2, 3, 4$, denote the 4-neighbors of $u^q_{i,j}$ in the qth channel; that is,

$$\begin{aligned} P^q_{1,(i,j)} &= u^q_{i,j+1}; \\ P^q_{2,(i,j)} &= u^q_{i,j-1}; \\ P^q_{3,(i,j)} &= u^q_{i+1,j}; \\ P^q_{4,(i,j)} &= u^q_{i-1,j}. \end{aligned} \tag{21}$$

Then, the explicit discrete schemes associated with (10), (11), and (12) are, respectively, as follows:

$$\begin{aligned} \left(u^q_{i,j}\right)^{n+1} &= \left(u^q_{i,j}\right)^{n} + (\Delta t) \\ &\cdot \sum_{k=1}^{4}\left[\left(C^q_{k,(i,j)}\right)^n\left(\left(P^q_{k,(i,j)}\right)^n - \left(u^q_{i,j}\right)^n\right)\right], \\ &\qquad \text{for } (i,j) \in \Theta^q, \end{aligned} \tag{22}$$

$$\begin{aligned} \left(u^q_{i,j}\right)^{n+1} &= \left(u^q_{i,j}\right)^{n} + (\Delta t) \\ &\cdot \sum_{k=1}^{4}\left[\left(C^q_{k,(i,j)}\right)^n\left(\left(P^q_{k,(i,j)}\right)^n - \left(u^q_{i,j}\right)^n\right)\right], \\ &\qquad \text{for } (i,j) \in \Theta^{q[v]}, \end{aligned} \tag{23}$$

$$\begin{aligned} \left(u^q_{i,j}\right)^{n+1} &= \left(u^q_{i,j}\right)^{n} + (\Delta t) \\ &\cdot \sum_{k=1}^{4} h^q(i,j)\left[\left(C^q_{k,(i,j)}\right)^n\left(\left(P^q_{k,(i,j)}\right)^n - \left(u^q_{i,j}\right)^n\right)\right], \\ &\qquad \text{for } (i,j) \in \Theta^{q[p]}. \end{aligned} \tag{24}$$

In these equations, Δt is the time step-size, and the superscript n denotes the nth iteration.

3. Experiments

In this section, three metrics used to evaluate denoising performance are first introduced containing Peak Signal to Noise Ratio (PSNR), Mean Structural Similarity Index Measure (MSSIM), and extended Feature Similarity index (FSIM_c), then the setting parameters are discussed, and finally the experimental results and comparisons are reported.

3.1. Three Metrics. PSNR, given in decibels (dB), is a measurement based on mean pixel intensity errors between the noise-free and the restored images. Higher PSNR means better denoising capability. Let u and $\hat{u}$ be the noise-free and the restored images, respectively, M denote the number of pixels in u, and $\|\cdot\|_{\mathrm{F}}$ denote the Frobenius norm; the calculation of PSNR is

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{255 \times 3M}{\sum_{q=1}^{3}\|u^q - \hat{u}^q\|_{\mathrm{F}}^2}\right). \tag{25}$$

MSSIM is a measure based on structural similarities [20]. It is bounded in the range $[0,1]$, where the closer the value is to 1, the better the denoising scheme is implied. MSSIM is mean value of the structural similarities between the blocks of the noise-free image u and the restored image $\hat{u}$. Therefore, the computation involves two blocks, denoted by y_1 and y_2, respectively. 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} the variances, and $\sigma_{y_1 y_2}$ the covariance, and let c_1 and c_2 denote two stabilization variables; then the calculation of SSIM is

$$\begin{aligned} \mathrm{MSSIM} &= \frac{1}{3M}\sum_{k}\mathrm{SSIM}(y_1, y_2) \\ \mathrm{SSIM}(y_1, y_2) &= \frac{\left(2\mu_{y_1}\mu_{y_2} + c_1\right)\left(2\sigma_{y_1 y_2} + c_2\right)}{\left(\mu_{y_1}^2 + \mu_{y_2}^2 + c_1\right)\left(\sigma_{y_1}^2 + \sigma_{y_2}^2 + c_2\right)}. \end{aligned} \tag{26}$$

FSIM_c is a measurement based on feature similarities [21]. Similar to MSSIM, it is also bounded in the range $[0,1]$, where the closer the value is to 1, the better the denoising performance is implied. This index is a combination of Phase Congruency (PC) and Gradient Magnitude (GM). Let $\mathbf{x}$ be a pixel and $\lambda > 0$ be a parameter. $S_L(\mathbf{x}) = S_{\mathrm{PC}}(\mathbf{x})S_G(\mathbf{x})$ and $S_C(\mathbf{x}) = S_I(\mathbf{x})S_Q(\mathbf{x})$, where symbol S means similarity, and the subscripts I and Q denote chromatic channels; the calculation of FSIM_c is

$$\mathrm{FSIM}_c = \frac{\sum_{\mathbf{x}\in\mathbf{\Omega}} S_L(\mathbf{x})\left[S_C(\mathbf{x})\right]^{\lambda}\mathrm{PC}_m(\mathbf{x})}{\sum_{\mathbf{x}\in\mathbf{\Omega}}\mathrm{PC}_m(\mathbf{x})}. \tag{27}$$

3.2. Parameter Setting

The Parameters in SPM. Five parameters are set in the proposed SPM method; (R, T, thr) are used in salt-and-pepper noise detection, and $(\Delta t, N)$ are used in diffusion operations. The parameters, T and thr, are two thresholds where the former judges the relationships between a component and its neighbors and the latter decides whether the component is corrupted or not. The influence of both on the PSNR denoising results for different intensity noise is shown in Figure 5. For all noise densities, R, T, thr, and Δt were set to 3, 25, 3, and 0.8, respectively. The parameter N is the desirable iterative time that depends on noise strength π and the time step-size Δt. Figure 6 illustrates that the PSNR and MSSIM results for fixed strength noise depend on the number of iterations when $\Delta t = 0.8$. From inset (a) in Figure 6, as the number of iterations n increases, the growth of PNSR tends to slow and ultimately achieves the stable value when n meets or exceeds the desirable number N, for any fixed noise. The same conclusion for the MSSIM measurement can be drawn

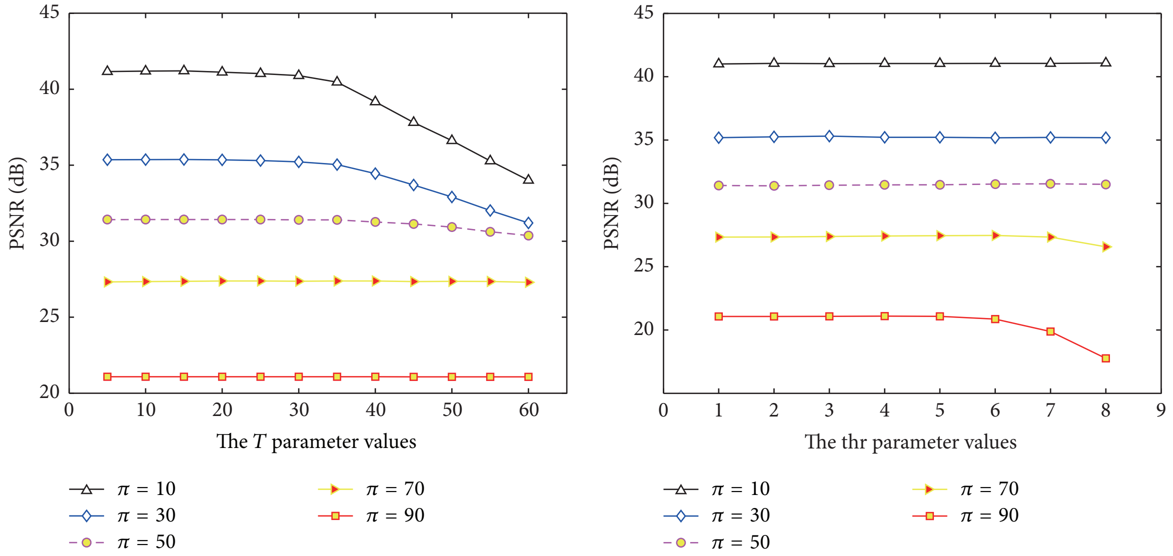


FIGURE 5: Dependence of PSNR measures on the parameters T and thr for various noise levels.

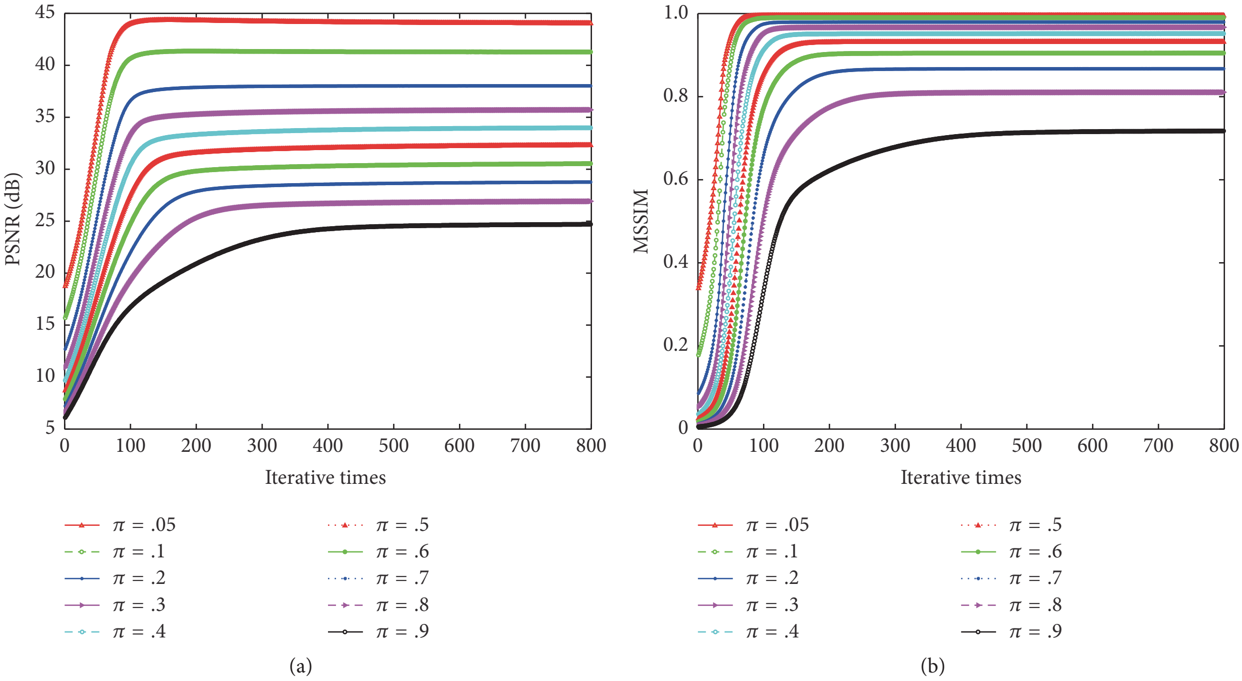


FIGURE 6: Illustration of the dependence of PSNR and MSSIM results on the number of iterations.

from inset (b) in Figure 6. In our experiments, the values of N are shown in Table 1.

The Parameters in RDM. Seven parameters are set in the proposed RDM method; (R, α, T_1, T_2) are used in random-valued impulse detection, and $(\Delta t, N^v, N^p)$ are used in diffusion operations. For all the noise densities, R, α, T_1, T_2, and Δt were set to 2, 14, 150, 320, and 0.5, respectively. The parameter N^v is the desirable number of iterations that is used in processing the corrupted pixels; the parameter N^p is the desirable number of iterations that is used in processing the possibly corrupted pixels. Both parameters depend on noise strength π and the time step-size Δt. When $\Delta t = 0.5$, the values of two parameters used in our experiments are shown in Table 2.

3.3. Experimental Results and Comparisons

Experimental Results and Comparisons in SPM. A test set, denoted by Γ, was built to validate the proposed SPM method. This set contained contaminative versions of seven images, and each image was corrupted by ten different densities of

Pepper 256 × 256 Baboon 512 × 512 Boat 256 × 256 Lena 256 × 256 London 512 × 512 Balloon 256 × 256 Voit 256 × 256

Figure 7: Seven noise-free images.

Table 1: The desirable number of iterations N for different density noise when $\Delta t = 0.8$.

π	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
N	150	190	270	380	510	620	690	750	810	850

Table 2: The value of the parameter N used in RDM.

π	5%	10%	20%	30%	40%	50%	60%
N^v	115	130	160	190	210	270	510
N^p	4	6	10	15	20	30	45

Table 3: Comparison of the efficiency with those of other methods for the Lena image in SPM.

Measure	Method	Salt-and-pepper noise π									
		5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
PSNR	MMF	28.81	28.09	25.57	21.90	18.09	14.60	11.86	9.61	7.82	6.37
	AMMF	28.81	28.09	25.03	22.09	21.11	19.35	18.47	16.91	15.04	11.30
	VMF	28.52	27.81	24.92	20.69	16.89	13.66	11.21	9.28	7.68	6.37
	TVL_1 [17]	26.37	26.31	26.15	26.00	25.81	25.45	25.09	23.94	17.89	12.40
	TVL_1 [18]	27.44	27.38	27.18	23.74	11.00	5.35	3.57	3.16	3.01	3.01
	Ours	**41.16**	**38.03**	**34.99**	**33.04**	**31.35**	**30.07**	**28.58**	**27.21**	**25.66**	**23.15**
MSSIM	MMF	0.83	0.82	0.78	0.66	0.45	0.25	0.13	0.06	0.03	0.01
	AMMF	0.83	0.82	0.75	0.69	0.68	0.63	0.61	0.54	0.45	0.21
	VMF	0.82	0.80	0.72	0.54	0.33	0.18	0.10	0.06	0.03	0.01
	TVL_1 [17]	0.72	0.72	0.72	0.71	0.70	0.69	0.68	0.64	0.42	0.18
	TVL_1 [18]	0.78	0.78	0.77	0.69	0.22	0.03	0.01	0.00	0.01	0.00
	Ours	**0.99**	**0.98**	**0.96**	**0.94**	**0.92**	**0.90**	**0.86**	**0.82**	**0.76**	**0.66**
FSIM_c	MMF	0.95	0.94	0.93	0.88	0.79	0.65	0.53	0.43	0.37	0.33
	AMMF	0.95	0.94	0.88	0.83	0.82	0.77	0.75	0.70	0.66	0.57
	VMF	0.94	0.94	0.90	0.84	0.73	0.61	0.50	0.42	0.37	0.33
	TVL_1 [17]	0.85	0.85	0.84	0.84	0.83	0.82	0.81	0.79	0.71	0.59
	TVL_1 [18]	0.87	0.87	0.87	0.83	0.49	0.33	0.30	0.29	0.29	0.29
	Ours	**1.00**	**1.00**	**0.99**	**0.98**	**0.97**	**0.96**	**0.95**	**0.92**	**0.89**	**0.82**

salt-and-pepper noise. The seven noise-free images are shown in Figure 7, taken from the CVG-Granada database. Using the values of parameters mentioned in SPM in Section 3.2, the proposed SPM method was applied to the test Γ. The PSNR, MSSIM, and FSIM results for the Lena image are reported in Table 3, and the visual results for Lena, Pepper, and Baboon are shown in Figures 8, 9, and 10, respectively.

To augment the evaluation, the proposed method is compared with other five methods including MMF [22], AMMF [23], VMF [1], TVL_1 in [18], and TVL_1 in [17]. The MMF filter used the sliding window of size 3 × 3. In the AMMF, when the noise densities were 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 0.6, 0.7, 0.8, and 0.9, the filtering window radius values were set to 1, 1, 2, 3, 3, 4, 4, 5, 6, and 7, respectively. The VMF used the sliding window of size 3 × 3. The source codes of TVL_1 in [18] and TVL_1 in [17] were taken from their authors, and the parameters used in the experiments were those recommended by the authors. The five methods were also applied to the same test set Γ. The PSNR, MSSIM, and FSIM results for the Lena image are also reported in Table 3. Visual comparison results for the Pepper and Baboon images are shown in Figures 9 and 10, respectively, and the information about these images is labeled sideways.

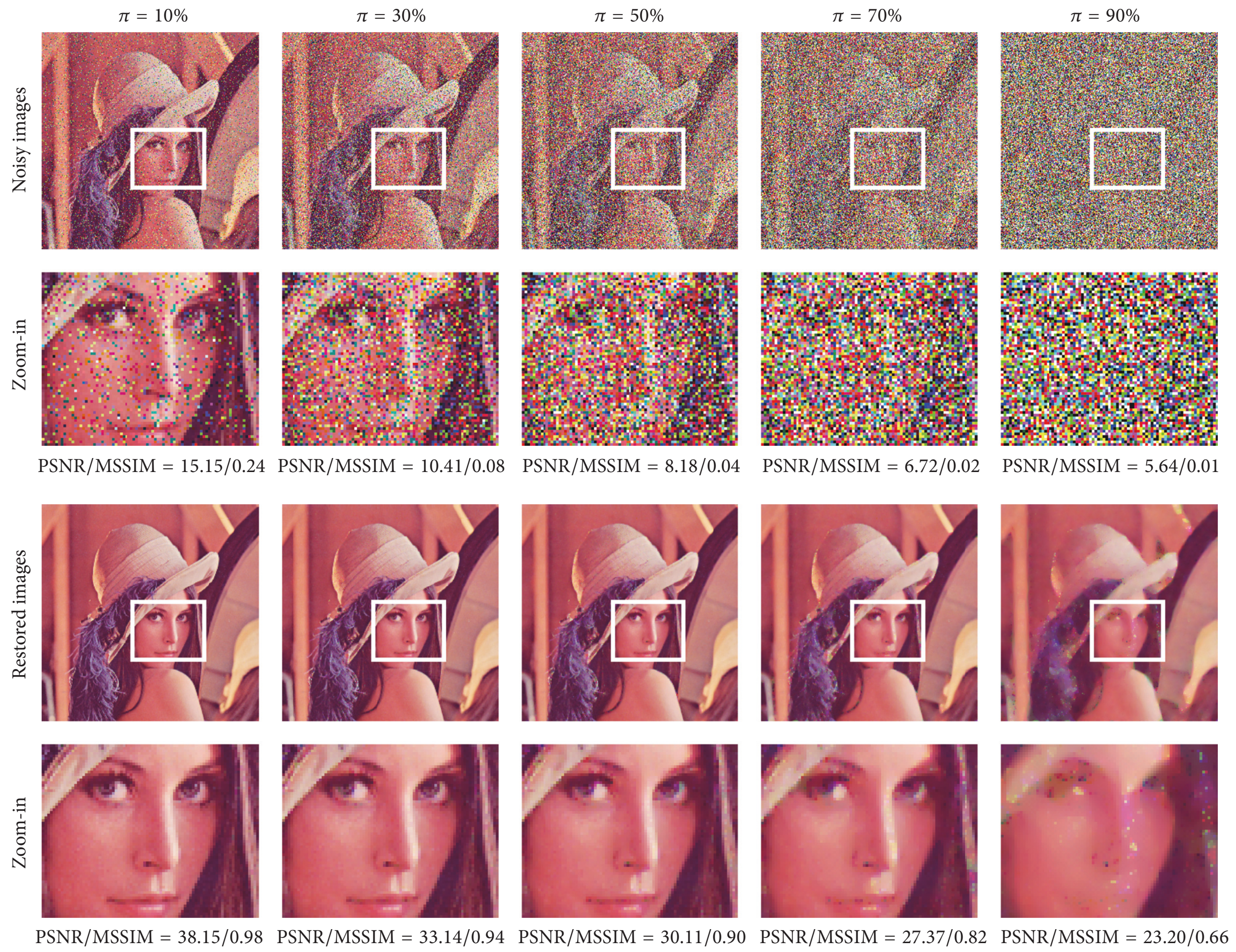


Figure 8: Visual results from the proposed SPM method for the Lena image.

In addition, the mean PSNR, MSSIM, and FSIM results for fixed noise are calculated for noisy images, the MMF, AMMF, VMF, TVL$_1$ in [18], TVL$_1$ in [17], and the proposed SPM method, respectively. The calculations are as follows:

$$\begin{aligned}\overline{\mathrm{PSNR}}_\pi &= \frac{1}{|\Gamma_\pi|}\sum_{k\in\Gamma_\pi}\mathrm{PSNR}_\pi(k),\\ \overline{\mathrm{MSSIM}}_\pi &= \frac{1}{|\Gamma_\pi|}\sum_{k\in\Gamma_\pi}\mathrm{MSSIM}_\pi(k),\\ \overline{\mathrm{FSIM}}_\pi &= \frac{1}{|\Gamma_\pi|}\sum_{k\in\Gamma_\pi}\mathrm{FSIM}_\pi(k).\end{aligned} \tag{28}$$

In the above three equations, Γ_π denotes the noisy images with the same density noise π in the test Γ and PSNR$_\pi(k)$, MSSIM$_\pi(k)$, and FSIM$_\pi(k)$ denote the PSNR, MSSIM, and FSIM values of the kth image with the π density noise, respectively. For example, if the MMF is used, $\overline{\mathrm{FSIM}}_{0.1}$ denotes the mean FSIM value of all the images with the 0.1 density noise that is associated with the MMF. The mean values of the three metrics for different methods on different noise densities are plotted in Figure 11.

Experimental Results and Comparisons in RDM. By using the seven noise-free images in Figure 7, another test set was built for the proposed RDM method, denoted by Ψ, where each noise-free image was corrupted by different seven random-valued impulses. Using the values of parameters mentioned in RDM in Section 3.2, the proposed RDM method was applied to the test Ψ. The PSNR, MSSIM, and FSIM results for the London image are reported in Table 4, and the visual results for the Pepper and Baboon images are shown in Figures 12 and 13, respectively.

For comparisons, other five methods, MMF, AMMF, VMF, AVMF [24], and TVL$_1$ [18], were applied on the test set Ψ. Both of the MMF and VMF used the sliding window of size 3 × 3. In both of the AMMF and AVMF, when the noise densities were 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6, the filtering window radius values were set to 1, 1, 2, 2, 3, 3, and 4, respectively. The source codes of TVL$_1$ [18] are taken from their authors, and the parameters used in the experiments are those recommended by the authors. The PSNR, MSSIM,

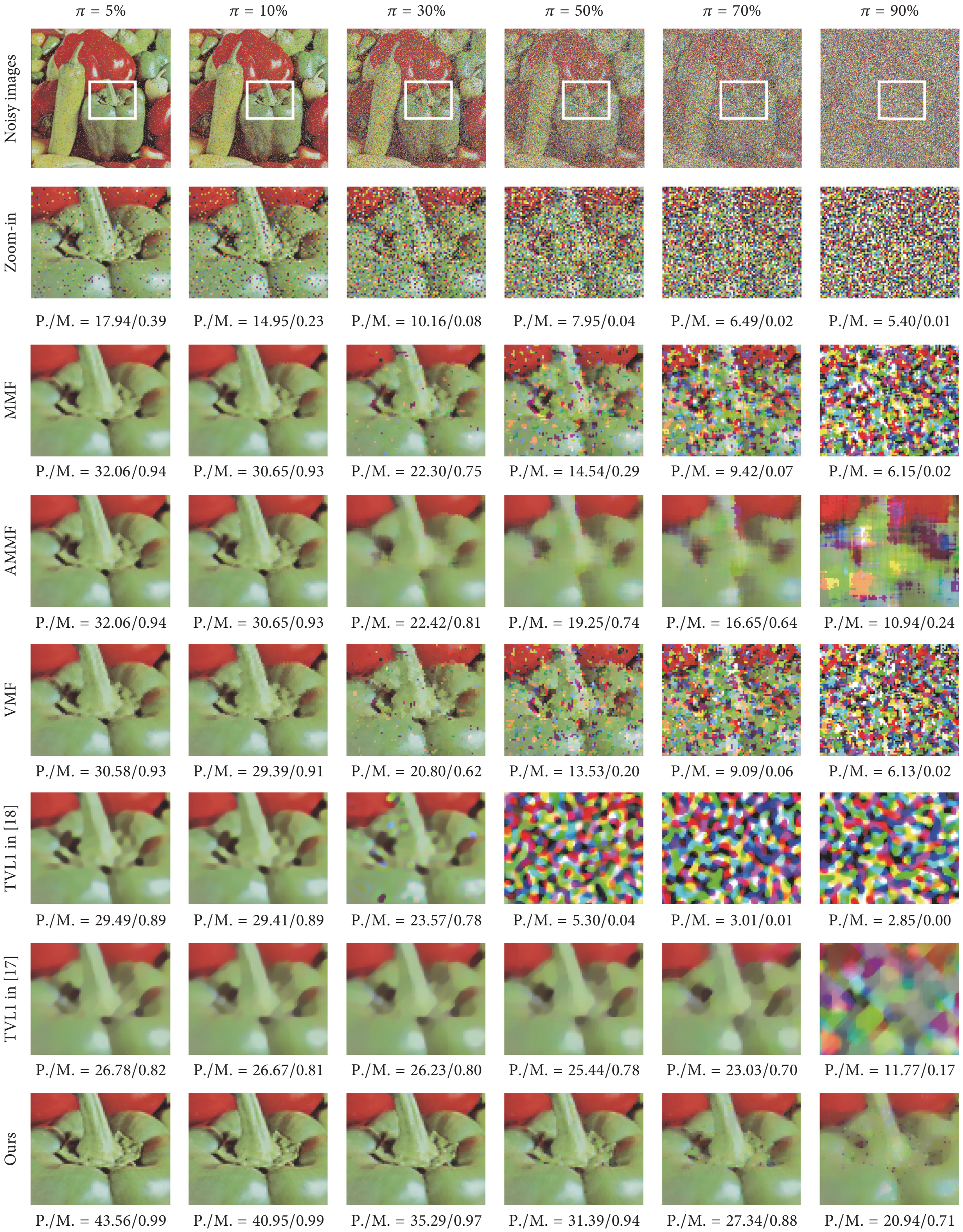


Figure 9: Visual zoom-in result comparison between different methods on the Pepper image.

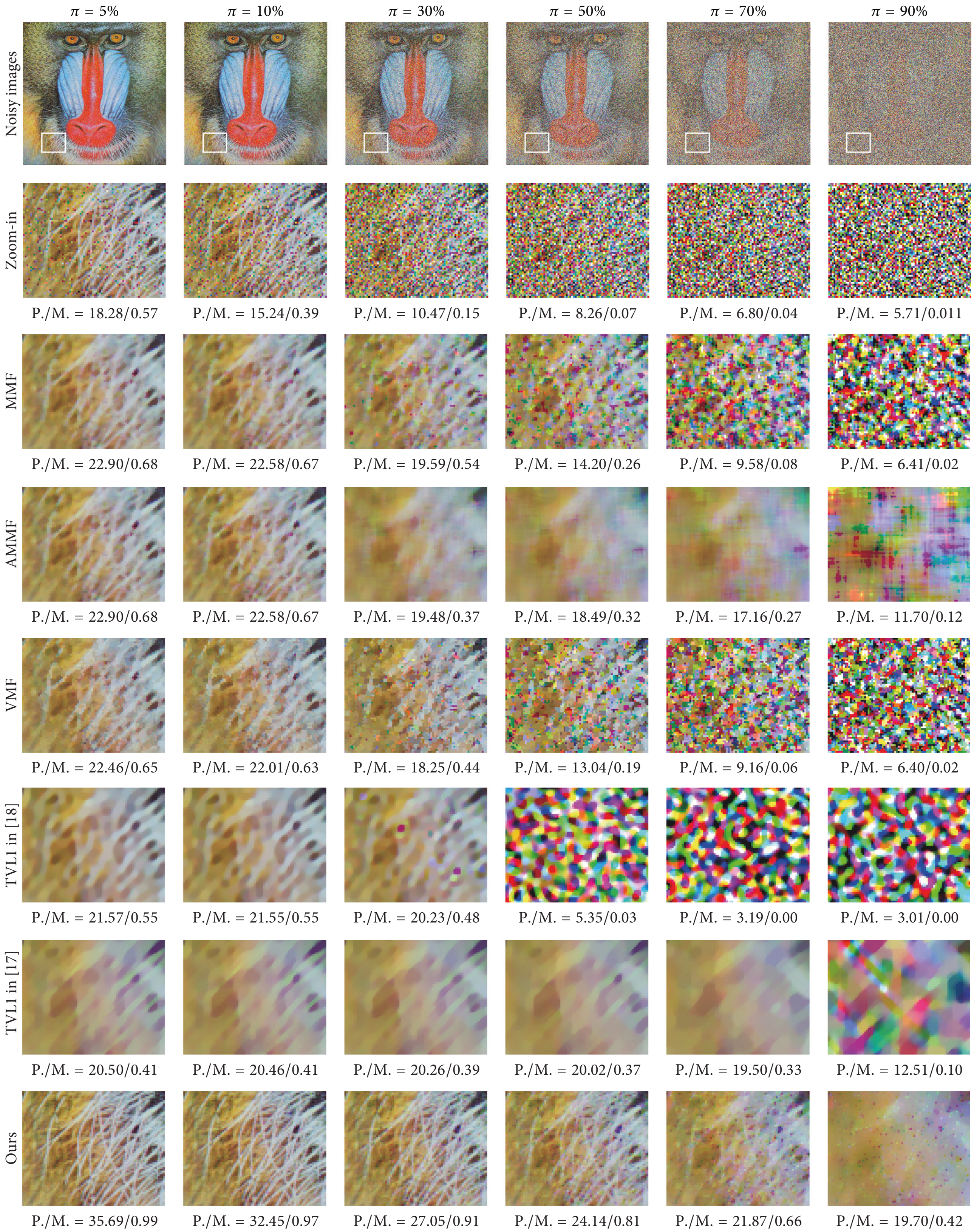


Figure 10: Visual zoom-in result comparison between different methods on the Baboon image.

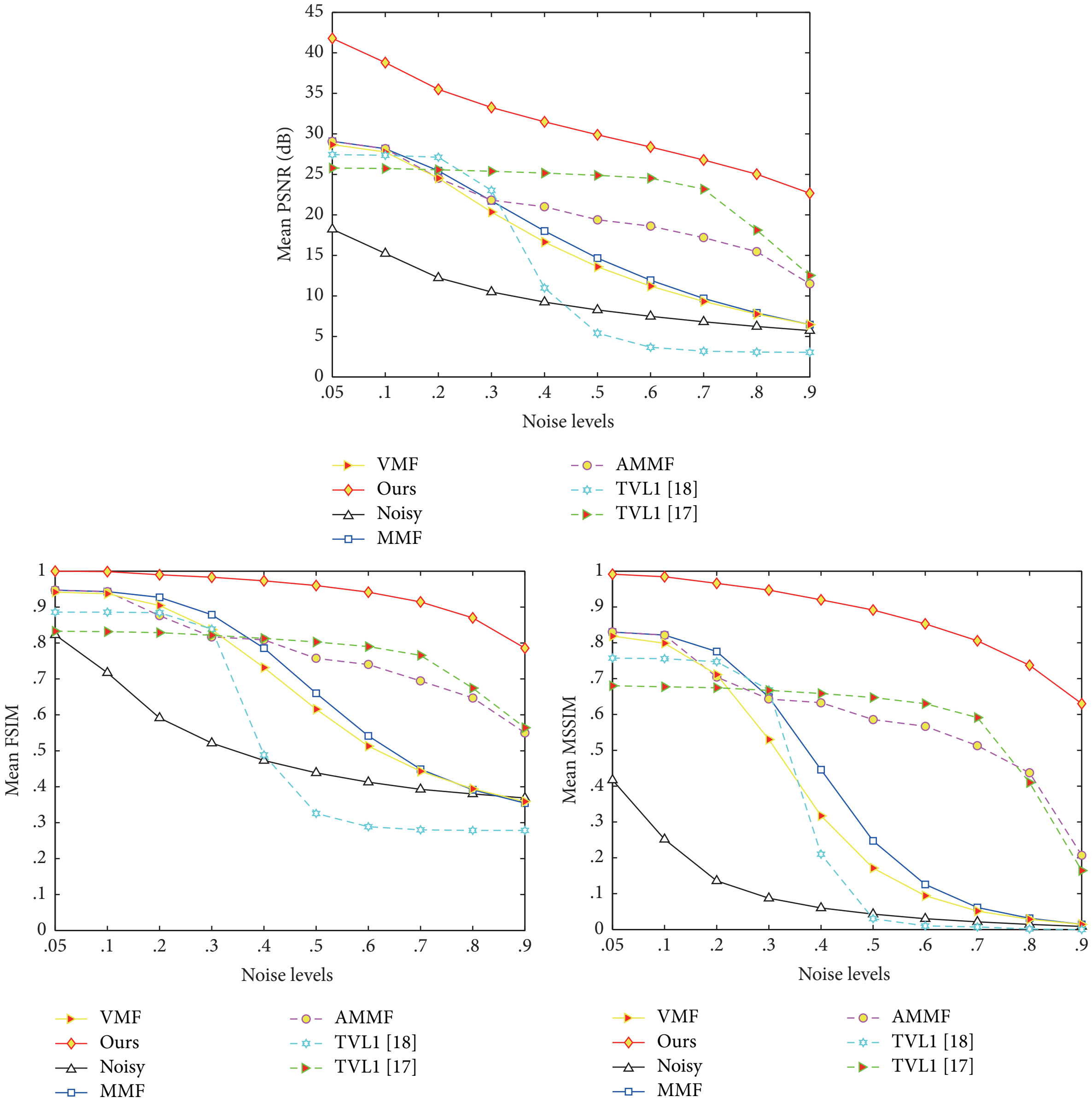


Figure 11: The mean PSNR/MSSIM/FSIM results from different methods in SPM.

and FSIM results for the London image are also reported in Table 4. Visual comparison results for the Pepper and Baboon images are also shown in Figures 12 and 13, respectively, and the information about these images is labeled sideways.

Similar to the comparisons in SPM, the mean PSNR, MSSIM, and FSIM results for fixed noise are calculated for noisy images, MMF, AMMF, VMF, AVMF, TVL_1 [18], and the proposed RDM method, respectively, according to (28). The mean values of the three metrics for different methods on different noise densities are plotted in Figure 14.

Observations and Conclusions. The following observations and conclusions can be drawn from the quantitative measurements and visual comparison results. First, the proposed method achieved the best results in every case tested. The proposed SPM method achieved 14.05 dB, 10.68 dB, 14.72 dB, 17.93 dB, and 8.27 dB improvements over the MMF, AMMF, VMF, TVL_1 [18], and TVL_1 [17] on average, respectively, in the PSNR results, and achieved 0.472, 0.278, 0.519, 0.554, and 0.293 improvements in the MSSIM results and achieved 0.254, 0.164, 0.274, 0.398, and 0.169 improvements in the FSIM results. The proposed RDM method outperformed MMF, AMMF, VMF, AVMF, and TVL_1 [18] by 3.56 dB, 4.16 dB, 4.80 dB, 3.73 dB, and 2.45 dB on average, respectively, in the PSNR results, and by 0.200, 0.182, 0.286, 0.231, and 0.135 in the MSSIM results and by 0.064, 0.097, 0.095, 0.102, and 0.071 in the FSIM results. Second, the proposed method has a strong capability to preserve detail. The proposed method reconstructed more image details from noisy images than the comparative methods. Third, the proposed method was more robust to different densities of noise than the comparative methods. In summary, the proposed method shows strong

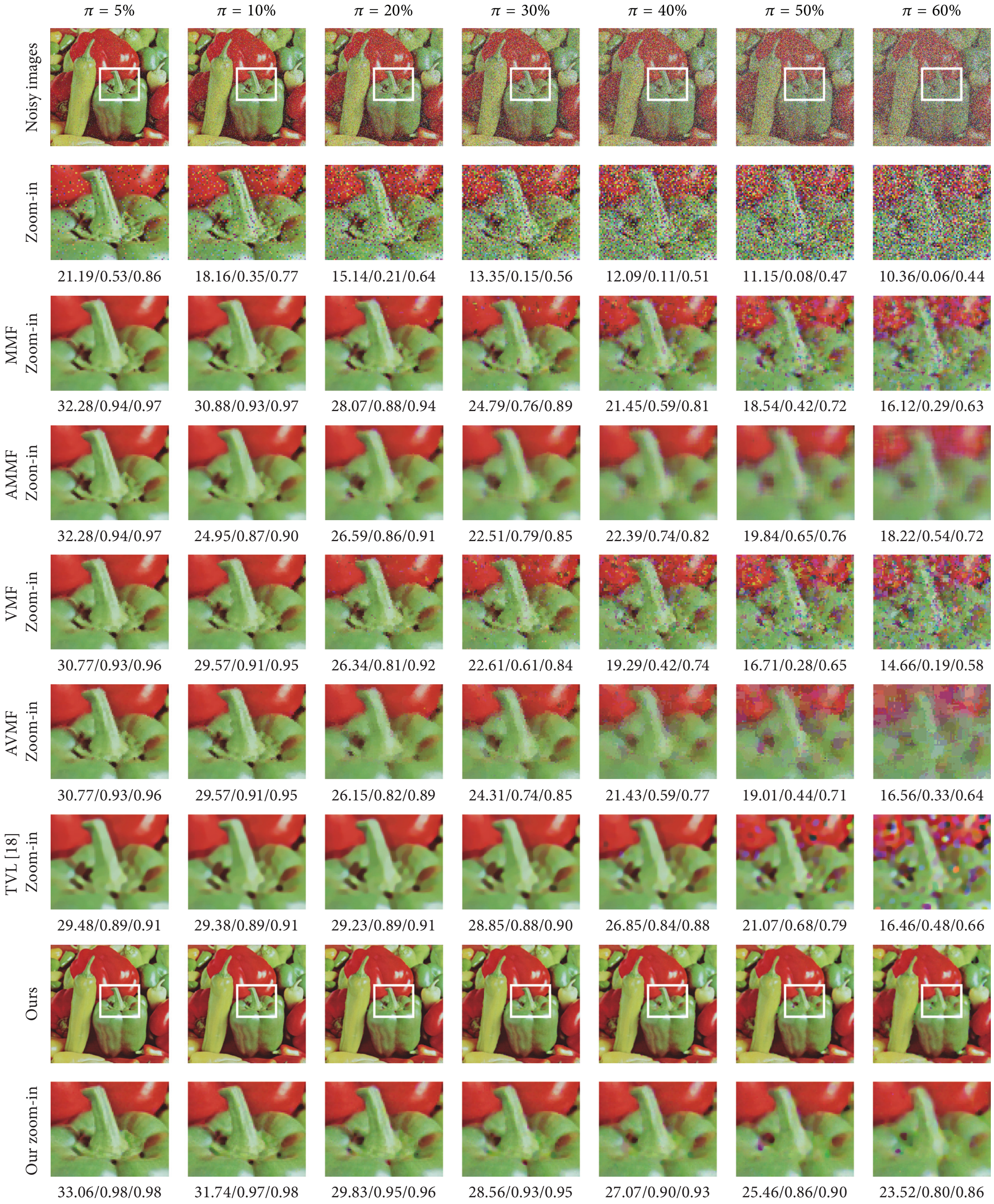


FIGURE 12: Visual results from different methods on the Pepper image in RDM. (PSNR/MSSIM/FSIM).

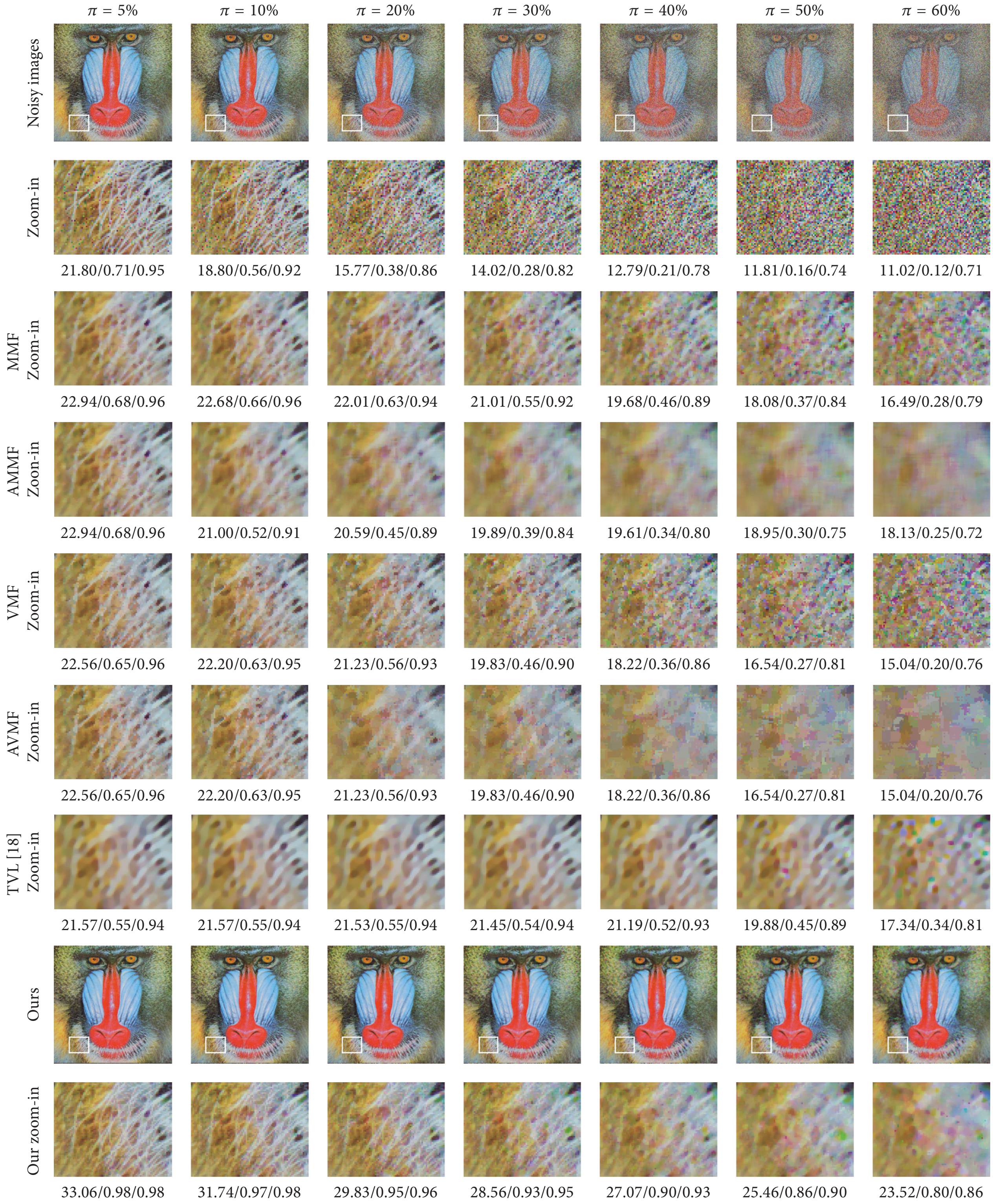


Figure 13: Visual results from different methods on the Baboon image in RDM (A/B/C = PSNR/MSSIM/FSIM).

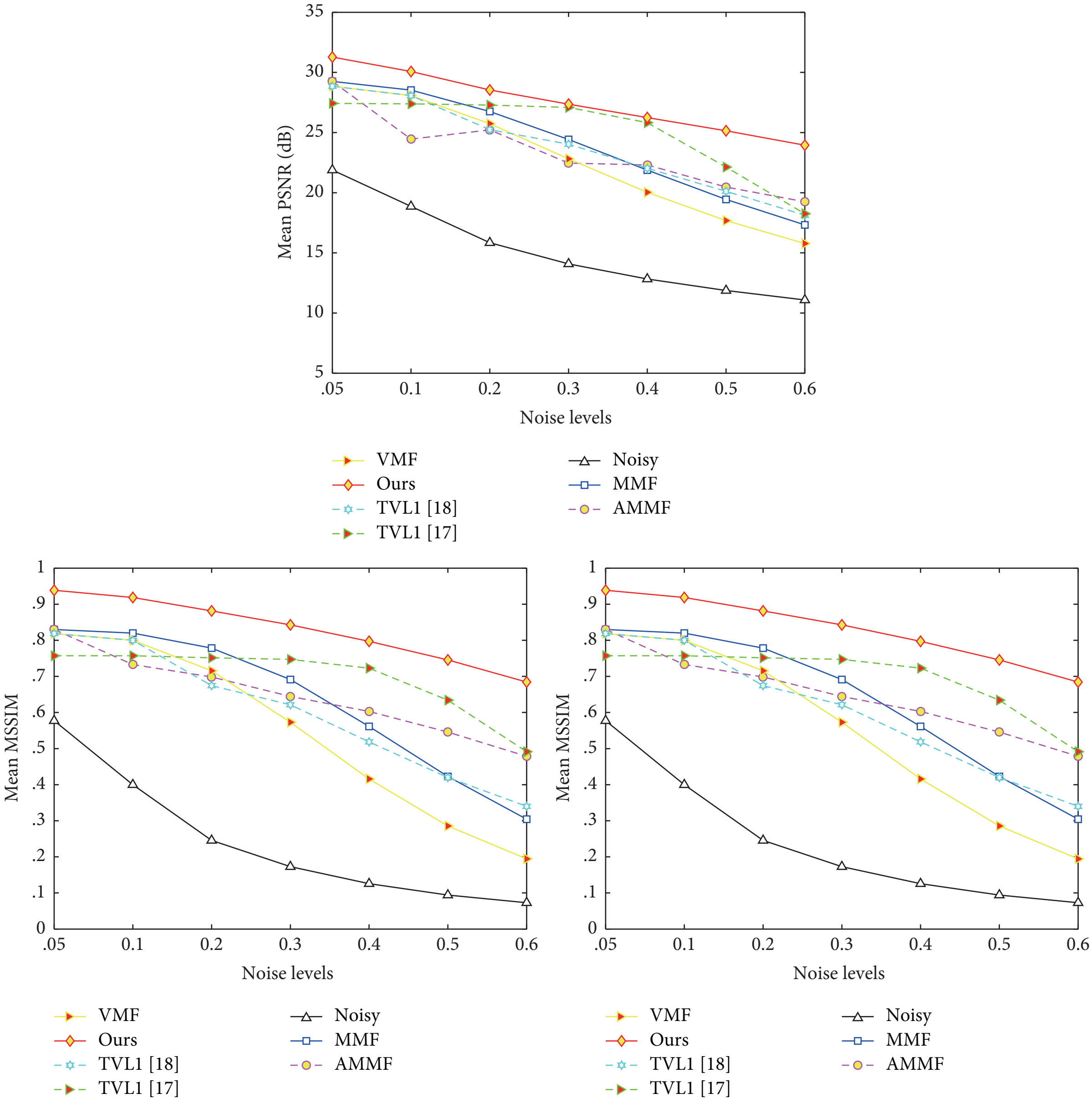


Figure 14: The mean PSNR/MSSIM/FSIM results from different methods in SPM.

capability of reducing noise not only in SPM but also in RDM, in terms of the PSNR/MSSIM/FSIM results, in addition to producing a higher visual perception quality in restored images, in comparison with results from other methods.

4. Conclusions

In this paper, a decision-based marginal total variation diffusion is proposed for impulsive noise removal. In contrast to vectorial methods, the proposed method only treats distorted components rather than distorted vectors. Furthermore, the proposed method divides components into different categories based on different noise characteristics. Components are divided into the corrupted and noise-free components in SPM and divided into the corrupted, noise-free, and the possibly corrupted components in RDM. Components falling into different categories are processed differently. If a component is corrupted, modified total variation diffusion is applied; if it is possibly corrupted, scaled total variation diffusion is applied; otherwise, the component is left unchanged. To achieve better results, a hierarchical scheme is adopted in RDM; the corrupted components are first processed and then the possibly corrupted ones. A total of 119 noisy images are tested. Experimental results show that the proposed method is robust to different noise strengths and suitable for different images, with strong noise removal capability as shown by PSNR/SSIM/FSIM results as well as the visual quality of restored images.

Table 4: Comparison of the efficiency with those of other methods for the London image in RDM.

Measure	Method	Random valued impulses π						
		5%	10%	20%	30%	40%	50%	60%
PSNR	MMF	33.70	32.92	30.89	27.86	24.71	21.72	19.29
	AMMF	33.70	28.33	28.89	26.01	25.57	23.67	22.21
	VMF	33.51	32.49	29.38	25.58	22.27	19.47	17.33
	AVMF	33.51	32.49	28.61	27.13	24.70	22.76	20.58
	TVL_1	31.07	31.00	30.87	30.63	29.50	25.38	20.49
	Ours	**37.56**	**35.82**	**33.78**	**32.03**	**30.60**	**29.06**	**27.56**
MSSIM	MMF	0.91	0.91	0.88	0.80	0.68	0.51	0.36
	AMMF	0.91	0.83	0.80	0.75	0.71	0.67	0.62
	VMF	0.91	0.90	0.82	0.67	0.49	0.32	0.21
	AVMF	0.91	0.90	0.79	0.74	0.66	0.59	0.53
	TVL_1	0.85	0.85	0.84	0.84	0.82	0.77	0.66
	Ours	**0.98**	**0.97**	**0.95**	**0.92**	**0.89**	**0.85**	**0.80**
FSIM_c	MMF	0.99	0.98	0.97	0.95	0.90	0.83	0.76
	AMMF	0.99	0.93	0.94	0.88	0.86	0.80	0.76
	VMF	0.98	0.98	0.96	0.92	0.86	0.78	0.71
	AVMF	0.98	0.98	0.93	0.91	0.83	0.79	0.72
	TVL_1	0.97	0.97	0.97	0.96	0.95	0.90	0.81
	Ours	**1.00**	**0.99**	**0.99**	**0.98**	**0.97**	**0.95**	**0.92**

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported by Foundation Science and Forefront Technology of Chongqing Science & Technology Commission under Grant no. cstc2016jcyjA0571 and also supported by Ph.D. Cultivation Foundation of Chongqing University of Posts and Telecommunications under Grant no. RC 2016002.

References

[1] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678–689, 1990.

[2] P. E. Trahanias, D. Karakos, and A. N. Venetsanopoulos, "Directional processing of color images: theory and experimental results," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 868–880, 1996.

[3] D. G. Karakos and P. E. Trahanias, "Generalized multichannel image–filtering structures," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 1038–1045, 1997.

[4] R. Lukac, B. Smolka, and K. N. Plataniotis, "Sharpening vector median filters," *Signal Processing*, vol. 87, no. 9, pp. 2085–2099, 2007.

[5] G. Peris-Fajarnes, B. Roig, and A. Vidal, "Rank–ordered differences statistic based switching vector filter," in *Image Analysis and Recognition*, A. Campilho and M. Kamel, Eds., vol. 4141 of *Lecture Notes in Computer Science*, pp. 74–81, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[6] B. Smolka, "Soft switching technique for impulsive noise removal in color images," in *Proceedings of the 5th International Conference on Computational Intelligence, Communication Systems, and Networks (CICSyN '13)*, pp. 222–227, IEEE, Madrid, Spain, June 2013.

[7] S. Morillas, V. Gregori, G. Peris-Fajarnés, and A. Sapena, "Local self–adaptive fuzzy filter for impulsive noise removal in color images," *Signal Processing*, vol. 88, no. 2, pp. 390–398, 2008.

[8] V. P. Ananthi and P. Balasubramaniam, "A new image denoising method using interval–valued intuitionistic fuzzy sets for the removal of impulse noise," *Signal Processing*, vol. 121, pp. 81–93, 2016.

[9] L. Jin, Z. Zhu, X. Xu, and X. Li, "Two–stage quaternion switching vector filter for color impulse noise removal," *Signal Processing*, vol. 128, pp. 171–185, 2016.

[10] L. Malinski and B. Smolka, "Fast adaptive switching technique of impulsive noise removal in color images," *Journal of Real-Time Image Processing*, pp. 1–22, 2016.

[11] S. Morillas, V. n. Gregori, and S. Antonio, "Fuzzy peer groups for reducing mixed Gaussian–impulse noise from color images," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1452–1466, 2009.

[12] L. Malinski and B. Smolka, "Fast averaging peer group filter for the impulsive noise removal in color images," *Journal of Real-Time Image Processing*, vol. 11, no. 3, pp. 427–444, 2016.

[13] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D. Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[14] S. Kim, "PDE-based image restoration: a hybrid model and color image denoising," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1163–1170, 2006.

[15] H. Deng, Q. Zhu, X. Song, and J. Tao, "A decision–based modified total variation diffusion method for impulse noise removal," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 2024396, pp. 1–20, 2017.

[16] J. C. Moreno, V. B. S. Prasath, and J. C. Neves, "Color image processing by vectorial total variation with gradient channels

coupling," *Inverse Problems and Imaging*, vol. 10, no. 2, pp. 461–497, 2016.

[17] J. Liu, T.-Z. Huang, X.-G. Lv, and J. Huang, "Restoration of blurred color images with impulse noise," *Computers and Mathematics with Applications*, vol. 70, no. 6, pp. 1255–1265, 2015.

[18] S. H. Chan, R. Khoshabeh, K. . Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3097–3111, 2011.

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

[20] 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.

[21] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: a feature similarity index for image quality assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011.

[22] A. Taguchi and D. Masayama, "The Relative Study of Vector Median Filters and Marginal Median Filters," *Ieice Technical Report Image Engineering*, vol. 100, pp. 27–33, 2000.

[23] S. Morillas, V. Gregori, and A. Sapena, "Adaptive marginal median filter for colour images," *Sensors*, vol. 11, no. 3, pp. 3205–3213, 2011.

[24] R. Lukac, "Adaptive vector median filtering," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1889–1899, 2003.

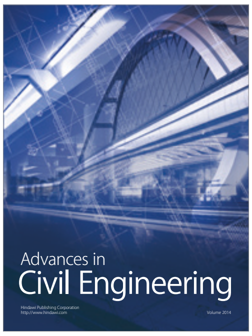
Journal of
Engineering

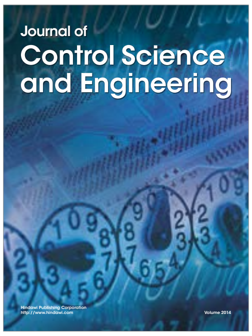
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

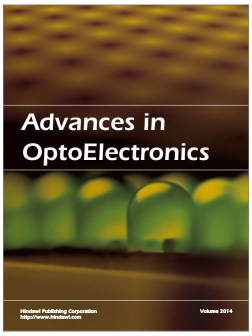
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

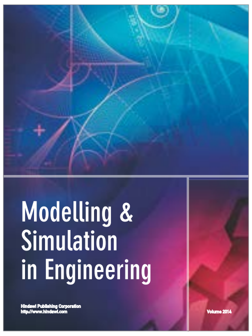
Journal of
Robotics

Journal of
Electrical and Computer Engineering

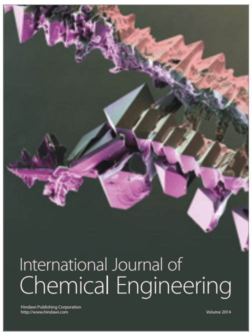

Advances in
OptoElectronics

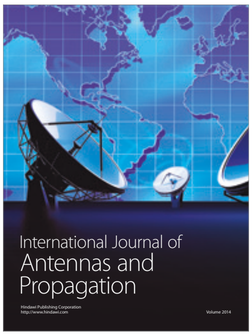
VLSI Design

International Journal of
Navigation and Observation

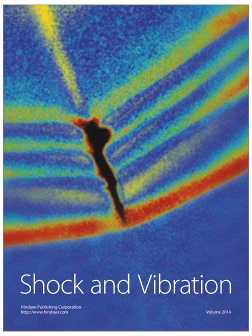

Modelling & Simulation in Engineering

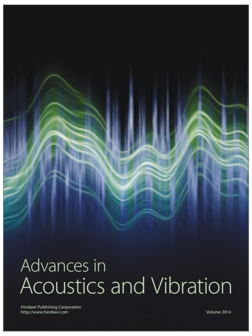
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration