Research Article

An Image Denoising Method Based on BM4D and GAN in 3D Shearlet Domain

Shengnan Zhang ¹, Lei Wang ¹, Chunhong Chang ¹, Cong Liu,¹ Longbo Zhang,¹
and Huanqing Cui ²

¹School of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China

²Shandong Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao 266590, China

Correspondence should be addressed to Lei Wang; wanglei0511@sdut.edu.cn and Huanqing Cui; cuihq@sdust.edu.cn

Received 14 January 2020; Revised 15 March 2020; Accepted 31 March 2020; Published 28 April 2020

Guest Editor: Chunjia Han

Copyright © 2020 Shengnan Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To overcome the disadvantages of the traditional block-matching-based image denoising method, an image denoising method based on block matching with 4D filtering (BM4D) in the 3D shearlet transform domain and a generative adversarial network is proposed. Firstly, the contaminated images are decomposed to get the shearlet coefficients; then, an improved 3D block-matching algorithm is proposed in the hard threshold and wiener filtering stage to get the latent clean images; the final clean images can be obtained by training the latent clean images via a generative adversarial network (GAN). Taking the peak signal-to-noise ratio (PSNR), structural similarity (SSIM for short) of image, and edge-preserving index (EPI for short) as the evaluation criteria, experimental results demonstrate that the proposed method can not only effectively remove image noise in high noisy environment, but also effectively improve the visual effect of the images.

1. Introduction

The typical image denoising methods can be commonly classified into three schemes: the filtering-based methods, the decomposition-based methods, and dictionary learning-based methods [1]. The classical filtering includes the median filter [2] and the wiener filter [3]. The basic principle of decomposition-based methods is to decompose the contaminated images into low-pass and high-pass subbands and then separate the image noise by manipulating the obtained coefficients. And, the wavelet transform is the typical decomposition tool. The basic principle of the dictionary learning-based methods is to sparsely represent the noisy image by overcomplete atoms, and only the large representation coefficients are used to reconstruct the original image while the small coefficients are discarded [4–6]. Another famous strategy for denoising is based on the self-similarity of the image, such as the block-matching and 3D filtering (BM3D) algorithm [7]. The above methods usually perform well, but they will lose the edge, texture, and other

details and result in blur and block effect when the noise is heavy.

Nowadays, since the adaptive space estimation strategy for nonlocal mean [8] can effectively alleviate the high complexity and low efficiency of nonlocal mean filtering, the nonlocal similarity becomes an effective feature for image denoising [9]. Its main disadvantage, however, is the loss of the directions. So, the geometric regularity of images cannot be effectively captured to sparsely represent the features of the original images [10]. Besides, the computation of the nonlocal similarity is implemented only in the spatial domain. As reported, the multiscale geometric transformations, such as contourlet transform, can greatly help to suppress the heavy noise in the frequency domain [11]. But this scheme is limited to the mathematical properties of the selected transformation. On the other hand, the deep learning technologies, such as the convolutional neural network [12] and the generative adversarial network [13], have achieved great success in the areas of image classification, target recognition [14], and image fusion [15].

Therefore, a novel image denoising method based on block matching with 4D filtering in the 3D shearlet transform domain and the generative adversarial network is proposed. The 3D shearlet transform provides better mathematical properties than the commonly used wavelet or contourlet to capture the anisotropic features of images in different scales and directions. In addition, the traditional BM3D is extended into four 4D space, which can effectively improve the edge and texture details of the images. The output of the BM4D is used as the input to the designed generative adversarial network to make full use of its good learning ability.

The remainder of this paper is organized as follows. The details of the proposed method are presented in Section 2. Experimental results and discussions are shown in Section 3. Finally, the whole paper is concluded in Section 4.

2. Methodology

2.1. The Architecture of Proposed Method. The overall structure of the proposed method is shown in Figure 1. Through multiscale decomposition and directional partition, the 3D shearlet coefficients are obtained. Then, the coefficients are input into the hard threshold and wiener filtering contained in the BM4D model. For the hard threshold, the similar cubes are extracted from volume observation and then they are combined together. If the distance between the cubes is smaller than the setting threshold, the collaborative filters are carried out to apply on the similar cubes. In the aggregation process, the basic volume estimation is generated by the adaptive convex combination. For the process of the wiener filtering, the aggregation is implemented by the inversion of the shearlet transformation. Finally, the clean results are obtained by the designed generative adversarial network.

2.2. The 3D Shearlet Transform. In the 3D space, the shearlet region P_i ($i = 1, 2, 3$) is obtained by combining the function system associated with the pyramid region in the 3D Fourier space R^3 .

For $d = 1, 2, 3, l = (l_1, l_2) \in Z^2$, the 3D shear system associated with P_d in the shearlet region is defined as a set:

$$\left\{ \psi_{j,l,k}^d = \left| \det A^{(d)} \right|^{j/2} \psi^{(d)} \left(B_l^{(d)} \left(A^{(d)} \right)^j x - k \right) : j \in Z, \right. \\ \left. l = (l_1, l_2) \in \Lambda_j, k \in Z^3, d = 1, 2, 3, \right\} \quad (1)$$

where $\psi^{(d)} \in L^2(R^3)$, $\{(l_1, l_2) : -2^j \leq l_1, l_2 \leq 2^j\}$, and $\Lambda_j \subset Z^2$. A^d is the specific anisotropic expansion matrix and B_l^d is the specific shear matrix. More details can be found in the literature [16, 17]. In Figure 2, a shearlet in 3D space is shown.

2.3. The Improved 3D Block-Matching Algorithm. Let $z: X \rightarrow R$ be the noise form:

$$z(x) = y(x) + \eta(x), \quad x \in X, \quad (2)$$

where y is the original unknown volume signal, $x \in Z^3$ is the 3D signal, and $\eta(\cdot) \sim N(0, \sigma^2)$ is an independent and

uniformly distributed Gaussian noise, whose standard deviation is noted as σ .

The goal of the improved 3D block-matching algorithm [18] was to obtain the estimation $\hat{y}$ of y from the noise observations. The implementation of the improved algorithm was divided into two cascade stages: the hard threshold and wiener filtering [19], each of which includes 3 steps: grouping, collaborative filtering, and aggregation.

For the hard threshold, let $C_{x_R}^z$ represent a cube whose volume is $L \times L \times L$, where $L \in \mathbb{N}$ was extracted from z . The similarity between two cubes is measured by photometric distance:

$$d(C_{x_i}^z, C_{x_j}^z) = \frac{\|C_{x_i}^z - C_{x_j}^z\|_2^2}{L^3}, \quad (3)$$

where $\|\cdot\|$ represents the summation of the squared differences between the corresponding intensities of the two cubes and the denominator L^3 is the normalization factor. No prefiltering is performed before cube matching, so the similarity of noise observations can be directly tested.

In the grouping step, cubes similar to each other are extracted from z and combined to form a group for each cube $C_{x_R}^z$. If the distance between two cubes was not larger than the predefined threshold $\tau_{\text{match}}^{\text{ht}}$, the two cubes are considered to be similar. Similar to $C_{x_R}^z$, we here firstly define a set that contains the indexes for the cubes as follows:

$$S_{x_R}^z = \{x_i \in X : d(C_{x_R}^z, C_{x_i}^z) \leq \tau_{\text{match}}^{\text{ht}}\}. \quad (4)$$

Then, a four-dimensional group is built by the above formula:

$$G_{S_{x_R}^z}^z = \prod_{x_i \in S_{x_R}^z} C_{x_i}^z, \quad (5)$$

where the reference cube (represented by R) matches a set of similar cubes located in the 3D data. Particularly, the coordinates x_R and x_i correspond to the tail and head of the arrow connecting the cubes in formula (4), respectively. Note that since the distance from any cube to itself is always 0, according to the definition of formula (4), each formula (5) must contain at least its reference cube.

In the collaborative filtering step, a joint four-dimensional transformation T_{4D}^{ht} was applied to each dimension of equation (5), respectively. Then, by a hard threshold operator γ^{ht} with the threshold $\sigma\lambda_{4D}$, the obtained four-dimensional group spectrum is

$$\gamma^{\text{ht}} \left(T_{4D}^{\text{ht}} \left(G_{S_{x_R}^z}^z \right) \right). \quad (6)$$

Representing the filter group, it is transformed into the following form:

$$T_{4D}^{\text{ht}^{-1}} \left(\gamma^{\text{ht}} \left(T_{4D}^{\text{ht}} \left(G_{S_{x_R}^z}^z \right) \right) \right) = \hat{G}_{S_{x_R}^z}^y = \prod_{x_i \in S_{x_R}^z} \hat{C}_{x_i}^y. \quad (7)$$

For each unknown volume data y , the estimated $\hat{C}_{x_i}^y$ of the original $C_{x_i}^y$ was extracted separately. Formula (7) was an overcompleted representation of the denoising data because

FIGURE 1: The architecture of the proposed image denoising method.

FIGURE 2: The shearlet in 3D space and its projection. (a) The 3D shearlet and (b) the top view of the 3D shearlet.

there may be overlap between the cubes in the same group and different groups.

In the aggregation step, the redundancy is used to generate a basic volume estimation by adaptive convex combination:

$$\hat{y} = \frac{\sum_{x_R \in X} \left(\sum_{x_i \in S_{x_R}^z} w_{x_R}^{ht} \hat{C}_{x_i}^y \right)}{\sum_{x_R \in X} \left(\sum_{x_i \in S_{x_R}^z} w_{x_R}^{ht} \chi_{x_i} \right)}, \quad (8)$$

where $w_{x_R}^{ht}$ is the group-related weight and $\chi_{x_i}: X \rightarrow \{0, 1\}$ is the feature (indicator) function of the $\hat{C}_{x_i}^y$ domain, that is, $\chi_{x_i} = 1$, at the coordinates of $\chi_{x_i} = 0$. The weight is defined as

$$w_{x_R}^{ht} = \frac{1}{\sigma^2 N_{x_R}^{ht}}, \quad (9)$$

where σ is the standard deviation of noise in z and $N_{x_R}^{ht}$ is the number of nonzero coefficients in formula (6). Since the coefficient always remains the same after doing the threshold operation, that is, $N_{x_R}^{ht} \geq 1$, the denominator of equation (9) is never equal to zero. The numerical $N_{x_R}^{ht}$ has two functions: on the one hand, it measures the sparsity of the threshold spectrum in (5) and on the other hand, it approximates the total residual noise variance of the group estimation in (6). As a result, the groups that are in a high degree of correlation will be given more weight, while other groups with larger residual noise are punished with less weight.

For the Wiener filtering, the cube matching is performed within the basic estimation of $\hat{y}^{ht}$. In fact, since the noise level in $\hat{y}^{ht}$ is much smaller than that in the noise observation z , it is expected to obtain the more reliable match to make the packet data more sparse. Formally, for each reference cube $\hat{C}_{x_R}^{ht}$ extracted from the basic estimated $\hat{y}^{ht}$, its cube-like coordinate set is constructed as follows:

$$S_{x_R}^{y,ht} = \left\{ x_i \in X: d\left(\hat{C}_{x_R}^{ht}, \hat{C}_{x_i}^{ht}\right) < \tau_{match}^{wie} \right\}. \quad (10)$$

The collaborative filtering here is implemented as an empirical Wiener filter. Similar to formula (6), it firstly uses the coordinate set (9) to extract a set of $G_{S_{x_R}^{y,ht}}^{y,ht}$ from $\hat{y}^{ht}$ and then defines the empirical Wiener filter coefficients as

$$W_{S_{x_R}^{y,ht}} = \frac{\left| T_{4D}^{wie} \left(G_{S_{x_R}^{y,ht}}^{y,ht} \right) \right|^2}{\left| T_{4D}^{wie} \left(G_{S_{x_R}^{y,ht}}^{y,ht} \right) \right|^2 + \sigma^2}, \quad (11)$$

where σ is the standard deviation of noise and T_{4D}^{wie} is a transformation operator that is composed of four one-dimensional linear transformations. Such transformations are usually different from those in T_{4D}^{ht} . Subsequently, the same

set of formula (10) is used to extract the second noise group from the observed z , noted as $G_{S_{x_R}^y}^z$. An element multiplication is implemented between the spectrum of the noise group and the wiener filter coefficient formula (11) as the coefficient shrinkage rate. The group's estimations are

$$\hat{G}_{S_{x_R}^y}^z = T_{4D}^{wice^{-1}} \left(W_{S_{x_R}^y} \cdot T_{4D}^{wic} \left(G_{S_{x_R}^y}^z \right) \right). \quad (12)$$

Then, the inversion of the 3D shearlet transform [14] is applied to shrink the spectrum. The final estimation of $\hat{y}^{wie}$ is generated by convex combination, which is similar to formula (8), and formula (4) is replaced by formula (10). The aggregation weight of the specific group estimation (11) is defined by the energy of the wiener filter coefficient (12):

$$w_{x_R}^{wie} = \sigma^{-2} \left\| W_{S_{x_R}^y} \right\|_2^2. \quad (13)$$

In this way, each formula (13) provides an estimation of the total residual noise variance of the corresponding formula (12).

2.4. The Generative Adversarial Network for Training. After obtaining the intermediate results, we can obtain the final denoising image by training the generative adversarial network (GAN) [20]. The training process is shown in Figure 3.

2.4.1. The Generator Network G . The generated network generates a fake image from the noise image, as shown in Figure 4. The generation network consists of 11 cascaded convolution layers, which are trained to learn the label image and the residual image of the input image. Internal connection is introduced into each block to save information and reduce the training time. In order to maintain good performance and reduce computational complexity, the network adopts a bottleneck structure, in which the number of the first feature mapping, the middle layer, and the last layers are 64 layers. According to suggestion from reference [21, 22], for low-level computer vision problems, a 3×3 convolution kernel is used in each convolution layer and the linear unit (ReLU) is used as the activation function.

2.4.2. The Discriminator Network D . As shown in Figure 5, the discriminator network D is trained to distinguish the fake image and the real image. It has four convolution blocks and two fully connected layers. Each convolution block consists of the convolution layer, the batch normalization layer, and the ReLU activation function. The size of the core K is 3×3, and the number of filters N increases from 64 to 256. The step size S of each convolution layer is 2 to reduce the resolution of the image. The probability that the inputting image is noiseless is generated by a fully connected layer of 1024 neurons.

2.4.3. Adversarial Training. The aim is to use the adversarial strategy to train a model to remove the image noising. Adversarial training is a way to train the generator network G to generate samples from real data $x \sim p_{\text{data}}$. The generator is input into a noise variable z with a distribution p_z and then trained to learn the mapping to the data space. The distribution of the generator model is

$$p_g \sim G(z; \theta_g), \quad (14)$$

where θ_g is the parameter of the generator network. When training a generator, the essentially exception is to maximize the probability that the samples match the data, which can be noted as $p_{\text{data}}(G(z; \theta_g))$.

To guarantee the above probability, the discriminator network D , whose input is the data sample x and the output is the $D(x, \theta_d)$, should learn to distinguish the generated samples from real samples. It must maximize the probability value assigned to the actual data samples and minimize the probability value assigned to the generated samples, that is,

$$\max_{\theta_d} E_{x \sim p_{\text{data}}} [\log(D(x))] + E_{z \sim p_z} [\log(1 - D(G(z)))]. \quad (15)$$

Both the generator and discriminator networks are alternately trained, and they try to cheat each other. Finally, when the generator has successfully learned how to generate the samples from p_{data} , the whole process is converged.

In Figure 6, the experiments on four groups of color images show the necessity and effectiveness of using the GAN.

3. Experimental Results and Discussions

In this section, two groups of experiments are implemented to show the performance of the proposed method. The platform is the Dell workstation M4800 with the Intel CPU 2.5 GHz and 32G RAM, operating under Matlab and Python. PSNR [23], SSIM [24], and the edge-preserving index (EPI for short) [25] are used as objective evaluation measurements. PSNR can be computed by the following formula:

$$\text{PSNR}(y, \hat{y}) = 10 \log_{10} \left(\frac{D^2 |\tilde{X}|}{\sum_{x \in \tilde{X}} (\hat{y}(x) - y(x))^2} \right), \quad (16)$$

where D is the peak of y , $\tilde{X} = \{x \in X: y(x) > 10 \cdot D/255\}$, and $|\tilde{X}|$ is the base of $\tilde{X}$.

The SSIM is defined as

$$\text{SSIM}(x, y) = l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma, \quad (17)$$

where $l(x, y)$, $c(x, y)$, and $s(x, y)$ can be computed as

$$\begin{aligned} l(x, y) &= \frac{2u_x u_y + c_1}{u_x^2 + u_y^2 + c_1}, \\ c(x, y) &= \frac{2\sigma_x \sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, \\ s(x, y) &= \frac{2\sigma_{xy} + c_3}{\sigma_{xy} + c_3}, \end{aligned} \quad (18)$$

FIGURE 3: Image denoising by training the GAN.

FIGURE 4: The structure of the generator network.

FIGURE 5: The structure of the discriminator network.

FIGURE 6: The compared experiments with and without GAN under the Gaussian noise of $\sigma = 25$.

in which x and y are the reference image and the image to be tested, respectively, u_x and u_y are the mean values of the two images, σ_x and σ_y are the standard deviation, σ_{xy} is the covariance of x and y , and c_1 , c_2 , and c_3 are the small constants whose values are positive. It is mainly to avoid the

instability when the denominator is 0 in the above formula. When $\alpha = \beta = \gamma = 1$ and $c_3 = c_2/2$, then we can get

$$\text{SSIM}(x, y) = \frac{(2u_x u_y + c_1)(2\sigma_x \sigma_y + c_2)}{(u_x^2 + u_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}. \quad (19)$$

FIGURE 7: The images used in the first experiment. (a) Cameraman, (b) Barbara, (c) man, (d) couple, (e) hill, (f) Lena, (g) house, (h) F16, (i) peppers, and (j) baboon.

TABLE 1: The results of the proposed method for Gaussian white noise with different standard deviations.

Image/PSNR	$\sigma = 5$	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$	$\sigma = 30$	$\sigma = 35$	$\sigma = 40$	$\sigma = 50$
Cameraman	39.10	34.79	33.01	31.43	30.12	29.37	28.69	27.80	26.78
Barbara	38.47	35.08	34.15	32.59	31.27	30.11	29.20	28.05	27.98
Man	38.01	35.49	32.76	31.59	30.30	29.12	28.93	27.81	26.82
Couple	38.58	35.39	33.62	31.36	30.45	29.73	28.99	28.40	27.01
Hill	38.47	34.88	32.90	31.63	30.81	30.16	29.59	28.77	27.96
Lena	38.81	35.99	33.94	33.23	32.27	31.59	30.91	30.11	29.88
House	39.02	36.47	35.17	33.97	33.20	32.68	32.09	31.12	30.79
F16	39.70	36.85	35.39	34.51	33.17	32.01	31.76	30.99	30.51
Peppers	37.39	34.28	33.01	31.91	31.67	30.73	30.88	29.70	29.15
Baboon	35.96	31.29	28.18	27.60	26.16	25.92	25.47	25.01	24.89

TABLE 2: The results of the different method for standard Gaussian white noise with $\sigma = 15$.

PSNR SSIM EPI Time/s	BM3D	EPLL	TNRD	WNNM	WSNM	Proposed
Cameraman	31.91	31.81	32.18	32.17	32.15	33.01
	0.72	0.71	0.76	0.80	0.79	0.86
	0.39	0.33	0.39	0.44	0.47	0.59
	0.51	29.86	0.88	136.75	172.14	3.51
Barbara	33.69	31.81	32.15	43.26	34.27	34.15
	0.77	0.75	0.77	0.83	0.87	0.86
	0.41	0.40	0.43	0.50	0.59	0.61
	3.05	178.25	1.64	825.50	986.36	7.62
Lena	34.94	32.91	33.44	33.54	33.53	33.94
	0.70	0.77	0.78	0.82	0.88	0.86
	0.40	0.41	0.43	0.48	0.55	0.59
	0.46	30.31	0.90	134.41	165.01	4.90
House	34.94	34.14	34.55	35.15	35.14	35.17
	0.69	0.78	0.81	0.81	0.83	0.87
	0.44	0.43	0.51	0.58	0.58	0.60
	0.78	32.32	0.87	136.16	170.27	4.74
Peppers	32.70	32.58	33.03	32.97	32.97	33.01
	0.66	0.81	0.83	0.84	0.85	0.84
	0.32	0.35	0.42	0.55	0.58	0.58
	0.46	29.55	0.91	132.65	160.62	3.89
Baboon	27.78	27.85	27.89	27.93	27.89	28.18
	0.64	0.77	0.79	0.80	0.84	0.84
	0.28	0.28	0.44	0.50	0.53	0.56
	1.49	111.98	1.22	490.38	654.31	7.11

TABLE 3: The results of the different method for standard Gaussian white noise with $\sigma = 25$.

PSNR SSIM EPI Time/s	BM3D	EPLL	TNRD	WNNM	WSNM	Proposed
Cameraman	29.45	29.23	29.71	29.63	29.70	30.12
	0.70	0.70	0.74	0.71	0.75	0.76
	0.33	0.31	0.35	0.45	0.45	0.50
	0.54	31.43	0.92	261.72	289.96	4.57
Barbara	31.18	28.98	29.63	31.83	31.84	31.27
	0.75	0.71	0.75	0.77	0.82	0.85
	0.37	0.33	0.40	0.44	0.56	0.60
	3.14	179.30	1.64	1706.84	2350.22	10.62
Lena	30.60	30.30	30.77	30.88	30.89	32.27
	0.68	0.72	0.75	0.80	0.80	0.82
	0.32	0.36	0.40	0.46	0.47	0.48
	0.47	34.78	0.91	259.41	287.63	6.22
House	32.86	32.04	32.54	33.23	33.23	33.28
	0.66	0.76	0.80	0.83	0.80	0.83
	0.29	0.40	0.50	0.56	0.56	0.57
	1.49	31.16	0.92	263.04	298.06	6.43
Peppers	30.16	30.07	30.55	30.40	30.41	31.67
	0.64	0.77	0.81	0.81	0.82	0.80
	0.26	0.31	0.40	0.40	0.44	0.55
	0.44	32.25	0.87	259.73	291.05	4.88
Baboon	25.23	25.34	25.39	25.42	25.44	26.16
	0.62	0.71	0.77	0.77	0.78	0.80
	0.26	0.25	0.34	0.46	0.46	0.50
	1.62	106.28	1.23	986.96	1296.27	9.60

FIGURE 8: The denoising results of different methods with the noise level $\sigma = 25$. (a) Noise image (20.17 db), (b) BM3D (25.23 db), (c) EPLL (25.34 db), (d) TNRD (25.39 db), (e) TNNM (25.42 db), (f) WSNM (25.44 db), and (g) proposed (26.16 db).

FIGURE 9: The results of removing the Gauss noise: from the left to right: the original image, the noisy image, and the denoising result.

FIGURE 10: The results of removing the Rician noise: from the left to right: the original image, the noisy image, and the denoising result.

Based on the edge contrast, the edge-preserving index (EPI) is defined as

$$EPI = \frac{[\sum |p_s(i, j) - p_s(i + 1, j)| + |p_s(i, j) - p_s(i, j + 1)|]}{[\sum |p_o(i, j) - p_o(i + 1, j)| + |p_o(i, j) - p_o(i, j + 1)|]}, \quad (20)$$

where $p_s(i, j)$ is a pixel from the testing image, $p_o(i, j)$ is a pixel from the original image, p_s and p_o are located in the edge region, i is the number of rows, and j is the number of columns. The range of EPI is 0 to 1. When EPI equals to 1, the edge of the image is completely maintained. When the EPI equals to 0, it means the image becomes a plane without any change. The larger the EPI value, the stronger the edge holding ability.

3.1. Experiment 1. Adding to the Gaussian white noise with different standard deviations, the first experiment is implemented under some popularly used natural images

shown in Figure 7. Five famous methods are used to be the comparative methods, including BM3D [26], EPLL [12], TNRD [27], WNNM [28], and WSNM [29] and the proposed method (proposed for short). All the results are reported in Tables 1–3, respectively.

In addition, the visual performance of different methods when the noise level $\sigma = 25$ is shown in Figure 8 to provide the subjective comparison. A small region, marked by the red rectangle, is enlarged to clearly display the visual difference.

3.2. Experiment 2. The second experiment is implemented under Gaussian noise and Rician noise. The testing volume data is a T1 brain network voxel with a size of $181 \times 217 \times 181$. The slice thickness is 1 mm. Without loss of generality, it is assumed that all the images are normalized to a real-valued signal in the intensity range $[0, 1]$. The experimental results are shown in Figures 9 and 10.

In Figures 9 and 10, the first row is one slice of the original volume data. In each column, it shows the lateral,

TABLE 4: The results of the different methods for Gaussian noise.

PSNR SSIM EPI	Filter	$\sigma/\%$									
		1	3	5	7	9	11	13	15	17	19
Gauss noise	Noisy data	40.00	30.46	26.02	23.10	20.91	19.17	17.72	16.48	15.39	14.42
		0.97	0.81	0.66	0.53	0.43	0.36	0.30	0.25	0.22	0.19
		0.20	0.19	0.17	0.17	0.14	0.13	0.10	0.09	0.06	0.04
	OB-NLM3D	42.47	37.57	34.73	32.82	31.42	30.32	29.40	28.61	27.91	27.28
		0.99	0.97	0.95	0.92	0.90	0.87	0.84	0.82	0.79	0.77
		0.29	0.28	0.26	0.25	0.23	0.20	0.19	0.17	0.15	0.14
	OB-NLM3D-WM	42.52	37.75	35.01	33.13	31.73	30.61	29.68	28.88	28.18	27.55
		0.99	0.97	0.95	0.93	0.90	0.88	0.85	0.83	0.80	0.78
		0.32	0.30	0.29	0.28	0.26	0.26	0.24	0.22	0.20	0.18
	ODCT3D	43.78	37.53	34.89	33.18	31.91	30.90	30.07	29.35	28.73	28.18
		0.99	0.97	0.95	0.93	0.91	0.89	0.88	0.86	0.85	0.83
		0.47	0.46	0.45	0.42	0.40	0.37	0.36	0.31	0.28	0.25
	PRI-NLM3D	44.04	38.26	35.51	33.67	32.37	31.29	30.40	29.65	28.99	28.40
		0.99	0.98	0.96	0.94	0.92	0.90	0.89	0.87	0.85	0.84
		0.66	0.64	0.62	0.60	0.59	0.56	0.55	0.55	0.53	0.51
	BM4D	44.09	38.39	35.95	34.38	33.21	32.28	31.50	30.82	30.23	29.70
		0.99	0.98	0.96	0.95	0.93	0.92	0.91	0.90	0.88	0.87
		0.70	0.69	0.68	0.67	0.60	0.58	0.56	0.56	0.55	0.54
	Proposed	45.08	39.31	36.79	35.12	33.79	33.31	32.16	31.58	30.79	30.53
		0.99	0.97	0.96	0.95	0.93	0.92	0.91	0.90	0.88	0.88
		0.75	0.72	0.68	0.67	0.63	0.61	0.60	0.59	0.57	0.55

TABLE 5: The results of the different methods for Rician noise.

PSNR SSIM EPI	Filter	$\sigma/\%$									
		1	3	5	7	9	11	13	15	17	19
Rician noise	Noisy data	40.00	30.49	26.09	23.20	21.04	19.32	17.88	16.65	15.57	14.60
		0.97	0.81	0.66	0.53	0.43	0.36	0.30	0.25	0.21	0.18
		0.20	0.19	0.17	0.17	0.14	0.13	0.10	0.09	0.06	0.04
	OB-NLM3D	42.41	37.45	34.54	32.51	30.97	29.71	28.62	27.64	26.74	25.91
		0.99	0.97	0.94	0.91	0.88	0.85	0.81	0.78	0.74	0.70
		0.29	0.28	0.26	0.25	0.23	0.21	0.19	0.17	0.15	0.14
	OB-NLM3D-WM	42.44	37.54	34.66	32.61	31.01	29.69	28.53	27.50	26.57	25.71
		0.99	0.97	0.95	0.92	0.88	0.85	0.81	0.77	0.74	0.70
		0.32	0.30	0.29	0.28	0.27	0.26	0.26	0.23	0.23	0.20
	ODCT3D	42.96	37.38	34.70	32.90	31.53	30.41	29.48	28.67	27.95	27.30
		0.99	0.97	0.95	0.93	0.90	0.88	0.86	0.84	0.82	0.80
		0.47	0.47	0.45	0.44	0.42	0.39	0.37	0.35	0.34	0.33
	PRI-NLM3D	43.97	38.19	35.34	33.37	31.94	30.74	29.75	28.88	28.10	27.39
		0.99	0.98	0.96	0.94	0.91	0.89	0.87	0.85	0.82	0.80
		0.66	0.65	0.63	0.60	0.57	0.55	0.54	0.52	0.48	0.46
	BM4D	44.08	38.34	35.83	34.17	32.89	31.82	30.90	30.06	29.29	28.57
		0.99	0.98	0.96	0.94	0.93	0.91	0.89	0.88	0.86	0.84
		0.71	0.70	0.69	0.67	0.62	0.60	0.59	0.57	0.56	0.55
	Proposed	45.10	39.49	36.97	35.38	33.61	32.92	32.18	31.61	30.42	29.71
		0.99	0.97	0.96	0.94	0.93	0.91	0.90	0.89	0.87	0.86
		0.76	0.71	0.68	0.67	0.64	0.63	0.62	0.60	0.58	0.57

coronal, sagittal, and cross results when the standard deviation is 0.05, 0.11, and 0.015, respectively. The results of under OB-NLM3D [30], OB-NLM3D-WM [31], ODCT3D [32], PRI-NLM3D [33], BM4D [34] and the proposed method are reported in Tables 4 and 5. All the quantitative results are shown in Tables 4 and 5, respectively.

By the comparison of all the above experimental results, it strongly demonstrates the efficiency of the proposed fusion methods, both visually and quantitatively. Specially, among the algorithms considered, the PSNR and SSIM

performance of the proposed method shows that the better results can be obtained, when the noise level increases. In addition, better visual effects can be obtained from the smoothness of flat areas, the preservation of details along the edges, and the accurate restoration of phantom intensity.

The time complexity of the proposed method is $O(N \log_2 N)$. The cost of computing time is indeed an annoying problem at present. The main reason is that the process of our method contains more steps than other methods. The good news is that our method is not the

slowest. We think it will be effectively solved in the future by improving the hardware conditions. In addition, some parallel computing methods (CUDA will be used in our plan) can also be used to reduce the time cost. This is also the work we will deal in the future.

4. Conclusion

In this paper, an image denoising method that is based on the BM4D in the 3D shearlet transform domain and GAN is proposed. The proposed method can make full use of the sparse representation of the 3D shearlet transform and the good deeply learning ability of the generative adversarial network. All the experimental results prove the effectiveness and accuracy of the proposed method in the form of subjective comparison and objective quantification, strongly demonstrating the superiority of the proposed method over the traditional image denoising methods.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by a Project of Shandong Province Higher Educational Science and Technology Program (J18KA362), the National Natural Science Foundation of China (61502282, 61902222), the Natural Science Foundation of Shandong Province (ZR2015FQ005), the Taishan Scholars Program of Shandong Province (tsqn201909109), and the Open Fund Project of Shandong Key Laboratory of Information Technology of Intelligent Mine in Shandong University of Science and Technology.

References

- [1] T. Yongpeng, J. Yu, and X. Cong, "Image denoising algorithm based on grouped dictionary and variational model," *Computer Application*, vol. 39, no. 2, pp. 551–555, 2019.
- [2] S. Yongkui, T. Wen, and C. Tongwen, "A method to remove chattering alarms using median filters," *ISA Transactions*, vol. 73, no. 1, pp. 201–207, 2018.
- [3] L. Zhiliang, S. Jia, G. Lihui et al., "Experimental study on active noise control based on wiener filtering," *Technology Wind*, vol. 5, no. 15, pp. 216–217, 2018.
- [4] M. Xiaole, H. Shaohai, and Y. Dongsheng, "SAR image denoising based on residual image fusion and sparse representation," *KSII Transactions on Internet & Information Systems*, vol. 13, no. 7, pp. 3620–3637, 2019.
- [5] H. Hao and L. Wu, "A new complex valued dictionary learning method for group-sparse representation," *Optik*, vol. 196, p. 163150, 2019.
- [6] L. Shuaiqi, L. Ming, L. Peifei et al., "SAR image denoising via sparse representation in shearlet domain based on continuous cycle spinning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2985–2992, 2017.
- [7] C. Bo, D. Ning, and B. Jing, "Application of BM3D denoising algorithms in instrument image recognition," *Zhejiang Electric Power*, vol. 38, no. 3, pp. 54–58, 2019.
- [8] Z. Wenwen and H. Yusheng, "Low rank sparse image denoising based on non-local self-similarity," *Computer Application*, vol. 38, no. 9, pp. 2696–2700+2746, 2018.
- [9] X. Su and Z. Yingyue, "Non-local mean denoising based on image segmentation," *Computer Application*, vol. 37, no. 7, pp. 2078–2083, 2017.
- [10] L. Yu and C. Sheng, "Summary of medical image segmentation methods," *Electronic Technology*, vol. 30, no. 8, pp. 169–172, 2017.
- [11] W. Rui and Z. Youchun, "Wavelet threshold denoising under new threshold function," *Computer Engineering and Application*, vol. 49, no. 15, pp. 215–218, 2013.
- [12] X. Wang, Q. Tao, L. Wang et al., "Deep convolutional architecture for natural image denoising," *International Conference Wireless Communications and Signal Processing*, vol. 5, no. 53, pp. 1–4, 2015.
- [13] Y. Min, W. Xintong, L. Yao et al., "Plausibility-promoting generative adversarial network for abstractive text summarization with multi-task constraint," *Information Sciences*, vol. 521, pp. 46–61, 2020.
- [14] P. Siddharth, R. Pranshu, and T. Jing, "An image augmentation approach using two-stage generative adversarial network for nuclei image segmentation," *Biomedical Signal Processing and Control*, vol. 57, p. 101782, 2020.
- [15] M. Jiayi, Y. Wei, L. Pengwei et al., "FusionGAN: a generative adversarial network for infrared and visible image fusion," *Information Fusion*, vol. 48, pp. 11–26, 2019.
- [16] B. Lei, Z. Xiongwei, Z. Yunfei et al., "Video saliency detection using 3D shearlet transform," *Multimedia Tools & Applications*, vol. 75, no. 13, pp. 7761–7778, 2016.
- [17] D. Labate and P. Negi, "3D discrete shearlet transform and video denoising," *Proceedings of Spie the International Society for Optical Engineering*, vol. 8138, no. 3, pp. 815–822, 2011.
- [18] D. Lijuan, W. Chunli, E. Qing et al., "Super-resolution algorithm of depth residual network image based on wavelet domain," *Journal of Software*, vol. 30, no. 4, pp. 941–953, 2019.
- [19] L. Sun, B. Jenbo, Y. Zheng et al., "A novel weighted cross total variation method for hyperspectral image mixed denoising," *IEEE Access*, vol. 6, no. 1, pp. 172–188, 2017.
- [20] C. Zailiang, Z. Ziyang, S. Hailan et al., "DN-GAN: Denoising generative adversarial networks for speckle noise reduction in optical coherence tomography images," *Biomedical Signal Processing and Control*, vol. 55, p. 101632, 2020.
- [21] Z. Kai, Z. Wangmeng, C. Yunjin et al., "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [22] L. Zhiyu, Z. Chengkun, and H. Min, "A nonsubsampling countourlet transform based CNN for real image denoising," *Signal Processing: Image Communication*, vol. 82, p. 115727, 2020.
- [23] F. Fan, Y. Ma, C. Li, X. Mei, J. Huang, and J. Ma, "Hyperspectral image denoising with superpixel segmentation and low-rank representation," *Information Sciences*, vol. 397–398, pp. 48–68, 2017.
- [24] M. Hongqiang, M. Shiping, X. Yuelel et al., "Image denoising based on improved stack sparse denoising self-encoder," *Computer Engineering and Application*, vol. 54, no. 4, pp. 199–204, 2018.

- [25] H. R. Shahdoosti and Z. Rahemi, "Edge-preserving image denoising using a deep convolutional neural network," *Signal Processing*, vol. 159, pp. 20–32, 2019.
- [26] X. Jia, Z. Junhua, and M. Lihui, "Denoising of salt and pepper noise based on improved BM3D algorithm," *Computer Engineering and Application*, vol. 54, no. 21, pp. 170–175, 2018.
- [27] H. Burger, C. Schuler, and S. Harmeling, *Learning How to Combine Internal and External Denoising Methods. Pattern Recognition*, Springer, Berlin, Germany, pp. 121–130, 2013.
- [28] G. Shuhang, Z. Lei, Z. Wangmeng et al., "Weighted nuclear norm minimization with application to image denoising," *Computer Vision and Pattern Recognition*, vol. 3, pp. 2862–2869, 2014.
- [29] X. Yuan, G. Shuhang, L. Yan et al., "Weighted schatten p-norm minimization for image denoising and background subtraction," *IEEE Trans. Image Process.* vol. 8, no. 10, pp. 4842–4857, 2016.
- [30] A. Welinton, S. Tiago, B. Gabriel et al., "Improving non-local video denoising with local binary patterns and image quantization," in *Proceedings of the 2016 29th SIBGRAPI Conference on Graphics*, pp. 1–9, Sao Paulo, Brazil, October 2016.
- [31] P. Coupé, P. Hellier, S. Prima, C. Kervrann, and C. Barillot, "3D wavelet subbands mixing for image denoising," *International Journal of Biomedical Imaging*, vol. 2008, Article ID 590183, 11 pages, 2008.
- [32] L. Xiangbo and Q. Tianshuang, "Denoise MRI images using sparse 3D transformation domain collaborative filtering," in *Proceedings of the 2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*, vol. 1, pp. 233–236, Shanghai, China, October 2011.
- [33] Y. Daitianyi, "Application of PRI_NLM3D denoising algorithms in CBCT 3D images," *New National Radiographic Digital Imaging and CT Technology*, vol. 10, no. 8, pp. 99–108, 2012.
- [34] X. Ping, C. Bingqiang, L. Xue et al., "A new MNF-BM4D denoising algorithm based on guided filtering for hyper-spectral images," *ISA Transactions*, vol. 2, no. 3, pp. 1–10, 2019.