International Journal of Innovative Computing, Information and Control
Volume **19**, Number **4**, August **2023**

ICIC International ©2023 ISSN 1349-4198
pp. **1103–1116**

BM3D WITH BI-HARD THRESHOLDING FOR IMAGE DENOISING IN HIGH-LEVEL NOISE SITUATIONS

Zekun Lv[1,2], Yang Geng[3], Hao Hou[4], Tao Lin[1,2], Xiaoya Dai[1,2] and Yingkun Hou[1,2,*]

[1]School of Information Science and Engineering
Guilin University of Technology
No. 319, Yanshan Street, Guilin 541006, P. R. China
lv1310671265@glut.edu.cn; {2281991672; 2531732885}@qq.com

[2]School of Information Science and Technology
Taishan University
No. 525, Dongyue Street, Taian 271000, P. R. China
*Corresponding author: ykhou@tsu.edu.cn

[3]Qingdao Academy of Chinese Medical Sciences
Shandong University of Traditional Chinese Medicine
No. 368, Hedong Road, Qingdao 266101, P. R. China
2022111566@sdutcm.edu.cn

[4]College of Intelligence and Information Engineering
Shandong University of Traditional Chinese Medicine
No. 4655, University Road, Jinan 250355, P. R. China
2020110512@sdutcm.edu.cn

Received December 2022; revised March 2023

Abstract. *The primary purpose of image denoising is to remove noise while better maintaining important detail information in the image. Block-matching and 3D fliteing (BM3D) is a state-of-the-art image denoising method. However, the intense isolated noise in the high-level noise situations is difficult to be removed due to the unique coefficient hard-thresholding in the original BM3D method. In this paper, we propose an improved BM3D algorithm using bi-hard thresholding, i.e., coefficient hard-thresholding and structural hard thresholding. The whole procedure consists of two significant steps: essential estimation and final estimation, and each step is divided into three minor steps: block-matching, collaborative filtering, and aggregation. In the basic estimation stage, we further enhance the sparsity of the transformed coefficients by reducing the coefficient hard-thresholding and adding structural hard-thresholding on the third dimension instead of the unique coefficient hard-thresholding in the original BM3D method. Finally, we compare the improved algorithm with the original BM3D method on the public dataset in high-level noise situations. Experimental results show that the proposed method improves the denoising performance.*

Keywords: Image denoising, Block-matching, Sparsity, Bi-hard thresholding

1. **Introduction.** As a low-level image processing task, image denoising has always been one of the hottest topics in computer vision. Since digital images are susceptible to in-noise contamination during the acquisition process, removing noise from the acquired images is a crucial step in image quality enhancement. In general, the purpose of image denoising is to recover the neat image $g(x, y)$ from the noisy image $f(x, y)$, i.e., $f(x, y) = g(x, y)+\eta(x, y)$, where $\eta(x, y)$ is the noise.

DOI: 10.24507/ijicic.19.04.1103

In the last decades, there have been many discussions on how to remove noise from images [1, 2, 3]. Non-local means (NLM) [4] is a milestone in the image-denoising field. The main idea of NLM is that the estimated value of the current pixel is obtained by the weighted average of its similar pixels in an image. Block-matching and 3D fliteing (BM3D) [5] combine spatial and transformation domains to enhance image sparsity. Weighted nuclear norm minimization (WNNM) [6] vectorizes similar blocks in the noisy image, stacks them into groups to transform them into a matrix, and then removes the noise by the singular value decomposition method. The non-local Haar (NLH) [7] proposed a similar-pixel grouping method with a double hard thresholding strategy on the Haar transform domain.

Deep learning has been widely used in image processing in the past few years as one of the most popular computer science techniques, for instance, DnCNN [8], FFDNet [9], CBDNet [10], NLNet [11], N3Net [12], MWCNN [13], Self-synaptic FHN [14], and AFF-YOLOv3 [15]. Most of these methods are based on convolutional neural networks and are trained on the training set to obtain the optimal model, and then the processing effect is verified on the test set.

BM3D has achieved satisfactory performance in image denoising and video denoising research, but the original BM3D method has failed to achieve its optimal performance in denoising at high-level noise situations. This paper uses a bi-hard thresholding strategy to improve the original BM3D. Compared with the BM3D method, a bi-hard thresholding strategy is used to improve the PSNR (peak signal-to-noise ratio) for better-denoised image results in high-level noise situations. Figure 1 shows the overall framework of the Bi-BM3D method for image denoising.

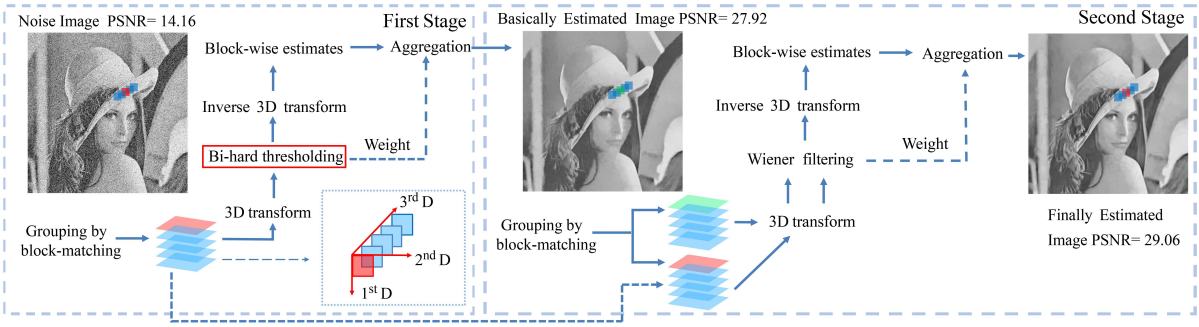


FIGURE 1. The overall framework of the proposed Bi-BM3D method for image denoising

The rest of this paper is organized as follows. In Section 2, we briefly survey the related work. In Section 3, we present the proposed bi-hard thresholding based BM3D method for image denoising. Extensive experiments are conducted in Section 4 and compared with the original BM3D image denoising method. The conclusion is given in Section 5.

2. Related Work.

2.1. **Conventional image denoising.** As Buades et al. proposed, NLM [4] estimates each pixel by calculating the weighted average of all similar pixels obtained by block-matching. All similar pixels are from the center of each similar image block. The similarity among the corresponding image patches determines the weights. Although NLM [4] is a milestone in the image-denoising field, it still tends to over smoothing image details when removing noises. BM3D [5] can be divided into three steps. First, each reference block is used to implement block-matching to obtain a similar block group. The separable 3D

transform on each block group is conducted, and hard thresholding filtering or Wiener filtering helps remove noise. Finally, the filtered results of each block group are aggregated to obtain the denoised image. Zhang and Ye [16] proposed an adaptive diffusion PDE model. According to the image edge index, different TV (total variation) models and Laplace equations are selected to reduce the step effect in the smooth area. In the NLH [7] method, block-matching is first performed to obtain similar image groups, then scan each image block, turning each image block into a column vector. All vectors are superimposed to obtain a 2D matrix. The row-matching is further performed to obtain similar pixel groups. Finally, a separable Haar transform is implemented on each similar pixel group to achieve signal decomposition.

2.2. **Deep learning-based image denoising.** DnCNN [8] employs convolutional neural networks and emphasizes the complementary roles of residual learning and batch normalization in signal recovery, which can lead to faster convergence and better performance despite deepening networks. NLNet [11] exploits the variation method of non-local self-similarity inherent in natural images and tries to combine it with deep neural networks. NLRN [17] is a combination of non-local and recurrent neural networks. A non-local recurrent network is proposed for image recovery tasks. Valsesia et al. [18] proposed a convolutional neural network that employs graph convolution layers to exploit local and non-local similarities. Zhang et al. [19] mainly proposed the network structure RDB (residual dense blocks), essentially the combination of residual and dense network structure. PRIDNet [20] proposed a novel pyramid real image denoising network to solve the blind denoising problem. Liang et al. [21] used the latest Transformer framework and presented a powerful baseline model based on the Swin Transformer (SwinIR) for image recovery.

Although these methods have performed well in image denoising, they rely on large datasets for model training, resulting in poor transferability. Also, they may require complex optimization algorithms to pursue excellent denoising effects. Therefore, we choose the simple and practical traditional image denoising method BM3D for image denoising. To improve the effect of denoising, we propose an improved algorithm that uses a dual hard thresholding strategy, namely coefficient hard thresholding and structural hard thresholding, to replace the original BM3D hard thresholding strategy.

3. Alogorithm.

3.1. First stage of denoising.

1) Block-grouping: Given a gray-scale noisy image $I \in \mathbb{R}^{h\times w}$, we select an image block with size $b^{ht} \times b^{ht}$ as the reference block R, and then search for the most similar blocks to R by calculating Euclidean distance d_l in the neighborhood around the reference block R by the step size as s^{ht}, where d_l is as the following:

$$d_l = \|Y_x - Y_R\|_2 \tag{1}$$

where Y_R is the reference block, and Y_x is a block the same size as Y_R at other locations in this neighborhood. Stack these similar blocks into a 3D array, and denote it as block group G_R^{ht}.

2) Collaborative bi-hard thresholding: To reduce the complexity of the calculation, we use separable 3D transforms for the block group, separate T_{3D}^{ht} into T_{2D}^{ht} in the first two dimensions and T_{1D}^{ht} in the third dimension. Generally, the bior1.5 wavelet transform is used for each image block to represent the information in the image blocks better. We use the simple Haar transform for the third-dimensional signals to better separate noise from the real signals.

After the separable 3D transform above, all of the coefficients whose magnitude is less than a given threshold are set to zeros by coefficient hard thresholding:

$$\Upsilon(x) = \begin{cases} 0, & \text{if } |x| \leq \lambda_{3\mathrm{D}}\sigma \\ x, & \text{otherwise} \end{cases} \tag{2}$$

where Υ is the coefficient hard thresholding results, x are the transformed coefficients, $\lambda_{3\mathrm{D}}$ is the hard thresholding parameter, and σ is the standard noise deviation.

It is different from the original BM3D method. We further introduce a structural hard-thresholding strategy to remove the intense isolated noise. The coefficient hard thresholding in the original BM3D method can only remove almost all the weak noise in high-level noise situations. Some intense isolated noises are difficult to be removed.

The block groups are denoted as G_{Υ}^{ht} after coefficient hard thresholding. According to the feature of the BM3D method, all transformed subbands from $n^{ht}/2$ to n^{ht} in G_{Υ}^{ht} are all high-frequency subband in which the transformed coefficients should largely be noises. To remove this noise in G_{Υ}^{ht} and make the transformed coefficients more sparse, we introduce a structural hard thresholding strategy and determine the degree of structural hard thresholding based on the comparison of the block-matching distance standard deviation *std* with a threshold M, where M is the empirical value obtained from the experiment, when $\sigma = 50, 75, 100$, the value of M is 3, 7, 15.

std is the standard deviation of the distance between the reference block and other similar blocks:

$$std = \sqrt{\frac{1}{n^{ht}-1}\sum_{q=2}^{n^{ht}} \left(d_l^q\right)^2} \tag{3}$$

where n^{ht} is the number of blocks, and d_l^q is the Euclidean distance between the reference block and other q-th similar blocks.

If the standard deviation of the block-matching distance is slight, and *std* is less than M, one can increase the strength of the structure hard thresholding to remove intense isolated noise better, that is, set all coefficients in the high-frequency band to 0. If the standard deviation of the block-matching distance is large, and *std* is more than M, the structural hard thresholding strategy is not performed. The specific algorithm steps are shown in Algorithm 1.

After collaborative filtering, if all blocks within a block group are sufficiently similar, their transform coefficients should also be very sparse. Therefore, we can assign weights according to the number of non-zero coefficients remaining after co-filtering:

$$w^{ht} = \begin{cases} \dfrac{1}{H}, & \text{if } H \geq 1 \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

where H is the number of retained (non-zero) coefficients after bi-hard thresholding. Finally, perform a separable 3D inverse transform and return these blocks to their original positions.

3) Aggregation: The weighted average result is the final result since there are overlapping image blocks when searching for similar blocks.

3.2. **Second stage of denoising.** The basic estimation image obtained in the first stage removed a lot of noise and some detailed information. Therefore, in the second stage, we use Wiener filtering for fine denoising to recover the detailed information and remove the noise.

Algorithm 1: Bi-hard thresholding

Input: Block groups $G^{ht}(i,j,k)$ after the 3D transform, the size of reference block is b^{ht}, the number of blocks is n^{ht}, λ_{3D} is the hard thresholding parameter, the standard deviation of distance for block-matching is std, the threshold is M.

```
// coefficient hard thresholding
for i ← 0 to b^ht do
    for j ← 0 to b^ht do
        for k ← 0 to n^ht do
            if G^ht(i,j,k) ≤ λ_3D σ then
                G^ht(i,j,k) ← 0
            else
                G^ht_Υ(i,j,k) ← G^ht(i,j,k)
// structural hard thresholding
for i ← 0 to b^ht do
    for j ← 0 to b^ht do
        if std ≤ M then
            for k ← n^ht/2 to n^ht do
                G^ht_Υ(i,j,k) ← 0
```

Output: Block groups after bi-hard thresholding processing G^{ht}_{Υ}

1) Block-grouping: Use block-matching in the basic estimate to find the locations of the most similar block to the reference block and record it. Using these locations, form two block groups, G^{wie}_n from the original noisy image and G^{wie}_b from the basic estimation image.
2) Collaborative Wiener filtering: We use a separable 3D transform on the two block groups, obtaining T^{wie}_b and T^{wie}_n. Compared with the double hard threshold filtering in the first stage, the Wiener filtering can suppress the noise in all frequency bands without completely removing the high-frequency information, which can achieve a better denoising effect. Therefore, we use the properties of Wiener filtering to recover some detailed information removed by the first stage of bi-hard thresholding. The shrinkage coefficient of the Wiener filter can be expressed as

$$W^{wie} = \frac{|T^{wie}_b|^2}{|T^{wie}_b|^2 + (\sigma)^2} \tag{5}$$

   In the second stage, the weights are calculated as follows:

$$w^{wie} = \left\|W^{wie}\right\|_2^{-2} \tag{6}$$

   where W^{wie} are the Wiener filter coefficients (5). Finally, the image block is separable 3D inverse transformed and returned to its original position.
3) Aggregation: The final estimate of the image is computed by aggregating all the obtained local estimates using a weighted average.

4. **Experiments and Results.** In this section, we describe the dataset and the implementation details and then quantitatively evaluate the proposed image-denoising algorithm. A desktop computer with an Inter Core CPU E5-2660v4 and a graphics card model NVIDIA TITAN RTX ran all code to investigate the proposed algorithm further.

4.1. **Datasets.** We evaluate our improved algorithm on four benchmark datasets on grayscale image denoising: the BM3D grayscale image dataset, RNI6 dataset, Set12, and BSD68 dataset.

The BM3D grayscale image dataset [5] is a collection of 11 images often used to evaluate image denoising methods. Moreover, the RNI6 dataset [22] is a collection of six images of different sizes.

The Set12 [23] dataset is a collection of 12 grayscale images of different scenes widely used to evaluate image denoising methods. The size of the image is 256×256 and 512×512.

The BSD [24] is a typical dataset for image denoising and super-resolution. Of the sub-datasets, BSD68 is a classical image dataset having 68 test images proposed by Martin et al. The dataset comprises various natural and object-specific images such as plants, people, and food.

4.2. **Experimental hyperparameter settings.** The detailed parameters of our experiments are shown in Table 1. We change one parameter at a time to assess its influence on Bi-BM3D. Table 2 lists the average PSNR results of Bi-BM3D with different parameter values on the RNI6 [22] dataset. It can be seen that: 1) The variations of PSNR results are from 0.01 dB (for the threshold M) to 0.04 dB (for the step size s^{ht}) when changing

Table 1. Image denoising parameters

		$\sigma \leq 40$	$\sigma > 40$
Parameters for Step 1 (ht)	T^{ht}_{2D}	2D-Bior1.5	2D-Bior1.5
	b^{ht}	8	8
	n^{ht}	16	16
	s^{ht}	3	2
	W	39	43
	λ_{3D}	2.7	2.8
Parameters for Step 2 ($Wiener$)	T^{w}_{2D}	2D-DCT	2D-DCT
	b^{w}	8	13
	n^{w}	32	32
	s^{w}	3	2
	W	39	51

Table 2. Average PSNR (dB) results of different parameters on the RNI6 dataset [22] images corrupted by AWGN (additive white Gaussian noise) noise ($\sigma = 50$). We change one parameter at a time to assess its influence on Bi-BM3D.

b^{w}	Value	8	9	11	13
	PSNR	25.26	25.27	25.27	25.28
s^{ht}	Value	2	3	4	5
	PSNR	25.28	25.27	25.25	25.24
W	Value	39	43	49	51
	PSNR	25.26	25.26	25.27	25.28
M	Value	1	2	3	4
	PSNR	25.27	25.27	25.28	25.27

individual parameters; 2) The performance on PSNR increases with increasing patch size b^w, window size W. For performance-speed tradeoff, we set $b^w = 13$, $W = 51$ in Bi-BM3D for efficient image denoising; 3) In the high-level noise situations, we have to use relatively large blocks to get a better grouping. All the parametric analyses demonstrate that Bi-BM3D is very robust in image denoising as long as the parameters are set in reasonable ranges.

4.3. **Results in BM3D images.** We first conducted experiments on the BM3D image dataset. We generate noisy synthetic images by adding AWGN with standard deviation σ ($\sigma \in \{50, 75, 100\}$) to the clean images. Figure 2 shows that the original BM3D adopts a

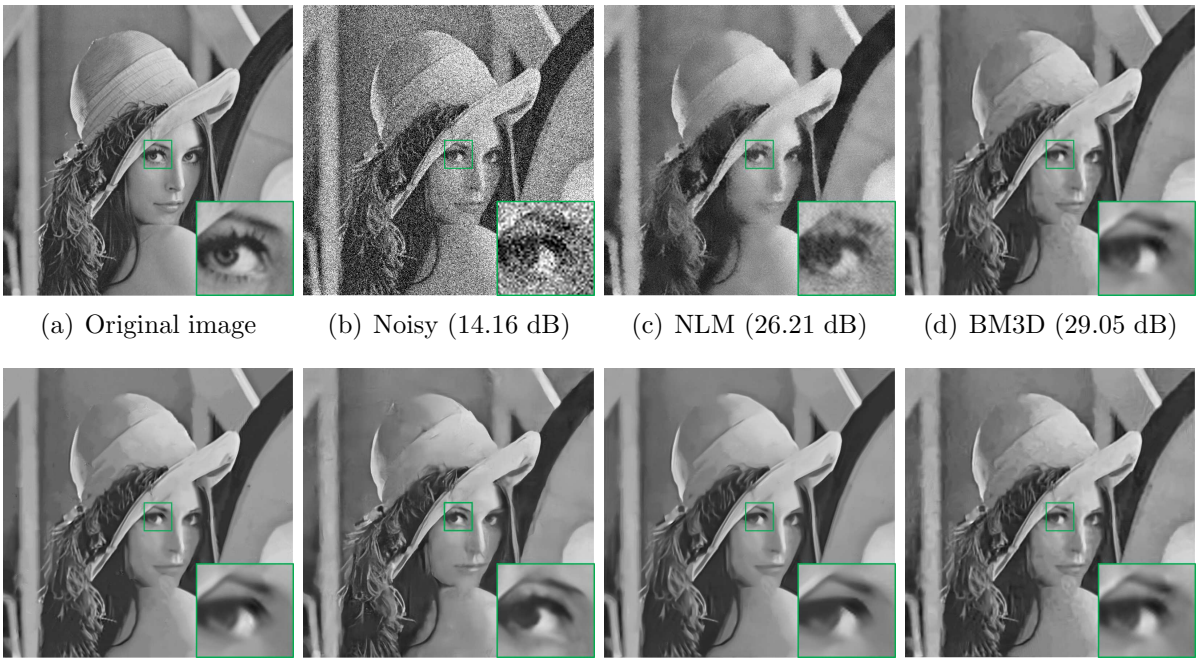
(a) (b) (c) (d)

FIGURE 2. Comparison of solitary noise removal between BM3D and Bi-BM3D (Noise level $\sigma = 50$): (a) BM3D (26.12 dB); (b) BM3D detail image; (c) Bi-BM3D (26.26 dB); (d) Bi-BM3D detail image

TABLE 3. Comparison of PSNR on BM3D grayscale image sets. The grayscale test sequences are corrupted by white Gaussian noise with different standard deviation values σ. For PSNR (dB): larger is better; best results are shown in bold font.

σ	Image Resolution	C.man 256^2	House 256^2	Peppers 256^2	Montage 256^2	Lena 512^2	Barbara 512^2	Boats 512^2	F.print 512^2	Man 512^2	Couple 512^2	Hill 512^2
50	NLM [4]	24.26	25.92	24.45	25.54	26.21	24.05	24.53	21.84	24.87	24.03	24.94
	BM3D [5]	26.12	29.69	26.68	27.90	29.05	27.23	26.78	24.53	26.81	26.46	27.19
	WNNM [6]	26.42	30.32	26.91	28.27	29.25	**27.79**	26.97	**24.67**	26.94	26.65	27.34
	DnCNN [8]	**26.91**	30.07	**27.31**	28.77	**29.39**	26.22	**27.11**	24.04	**27.13**	**26.84**	**27.37**
	NLH [7]	26.50	**30.56**	27.06	**29.18**	29.21	27.47	26.91	23.46	26.88	26.58	27.34
	Bi-BM3D	26.26	29.76	26.80	27.81	29.06	27.27	26.82	24.55	26.81	26.52	27.24
75	NLM [4]	21.84	23.20	21.76	22.21	23.75	21.92	22.48	19.18	22.83	22.16	23.08
	BM3D [5]	24.33	27.51	24.73	25.52	27.26	25.12	25.12	22.83	25.32	24.70	25.68
	DnCNN [8]	–	–	–	–	–	–	–	–	–	–	–
	WNNM [6]	24.55	28.25	24.93	25.73	27.54	**25.81**	**25.29**	**23.02**	25.42	**24.85**	25.88
	NLH [7]	**24.57**	**28.56**	**25.12**	**26.56**	**27.59**	25.61	25.25	22.19	**25.44**	**24.85**	**25.89**
	Bi-BM3D	24.35	27.53	24.82	25.36	27.27	25.23	25.16	22.87	25.34	24.77	25.74
100	NLM [4]	20.17	21.43	20.01	20.24	22.00	20.54	21.04	17.73	21.35	20.84	21.68
	BM3D [5]	23.07	25.87	23.39	23.89	25.95	23.62	23.97	21.61	24.22	23.51	24.58
	DnCNN [8]	–	–	–	–	–	–	–	–	–	–	–
	WNNM [6]	**23.56**	26.68	23.46	24.16	26.20	24.37	24.10	**21.81**	24.36	23.55	**24.75**
	NLH [7]	23.33	**27.09**	**23.80**	**24.52**	**26.45**	**24.43**	**24.14**	21.39	**24.45**	**23.69**	24.74
	Bi-BM3D	23.09	25.99	23.39	23.76	25.99	23.78	24.02	21.70	24.26	23.61	24.68

coefficient hard thresholding strategy, which is difficult to remove for some intense isolated noises. So we use a bi-hard thresholding strategy to remove this noise and preserve the details of the image as much as possible. Table 3 gives a quantitative comparison of our proposed improved algorithm with other denoising methods. At noise level $\sigma = 50$, our method is about 0.2 dB lower than the average of deep learning-based DnCNN [8]. The non-local methods WNNM [6] and NLH [7] are better than ours at all noise levels. We visualize the denoising results of different algorithms in Figure 3. Compared with NLM [4] and BM3D [5], our proposed method achieves better PSNR values in high-level noise situations, and we retain more detailed information and remove intense isolated noise, resulting in better visual effects.



(a) Original image (b) Noisy (14.16 dB) (c) NLM (26.21 dB) (d) BM3D (29.05 dB)

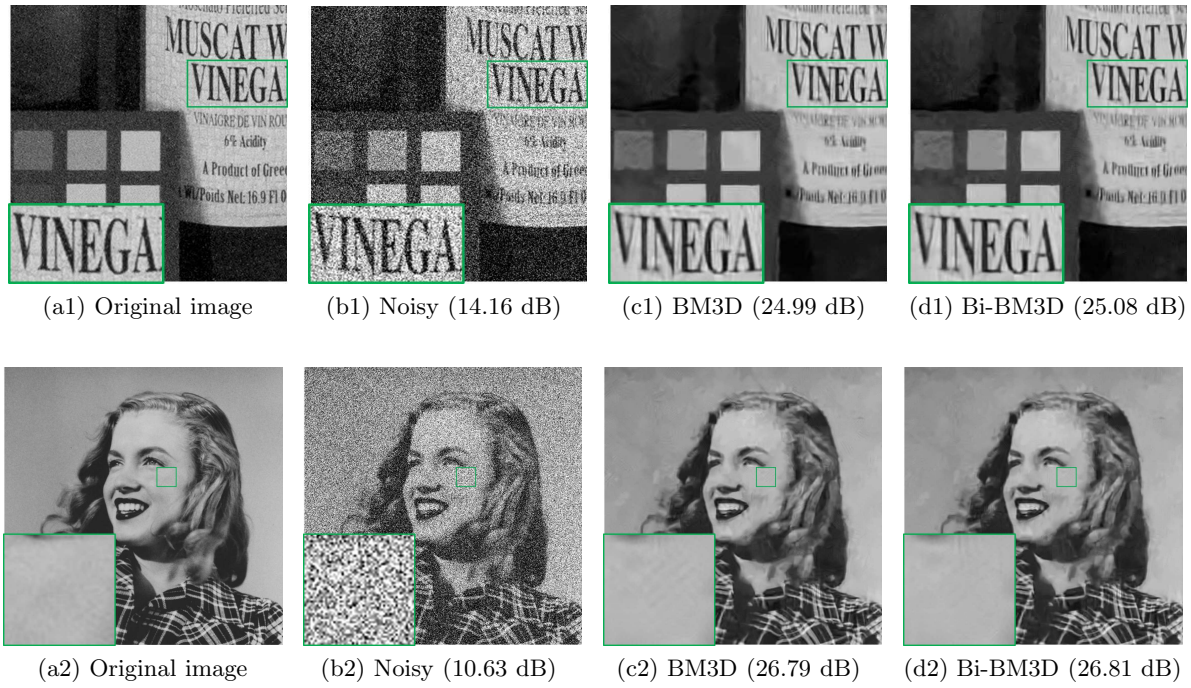
(e) WNNM (29.25 dB) (f) DnCNN (29.39 dB) (g) NLH (29.21 dB) (h) Bi-BM3D (29.06 dB)

FIGURE 3. Average PSNR (dB) results of different methods on BM3D images corrupted by AWGN noise ($\sigma = 50$)

4.4. **Results in RNI6.** We conducted comparative experiments with the original BM3D algorithm on the RNI6 dataset [22]. Table 4 and Figure 4 show the denoising results for images with high noise intensity. The original BM3D algorithm was unable to remove

TABLE 4. PSNR results of BM3D and Bi-BM3D on the RNI6 [22] benchmark dataset. For PSNR (dB): larger is better; best results are shown in bold font.

σ	Image	Building	Chupa_Chups	David_Hilbert	Marilyn	Old_Tom_Morris	Vinegar	Average
50	BM3D [5]	23.86	**26.42**	26.26	28.85	21.12	24.99	25.25
	Bi-BM3D	**23.87**	26.40	**26.31**	**28.87**	**21.15**	**25.08**	**25.28**
75	BM3D [5]	22.77	**24.55**	24.82	26.79	19.62	23.74	23.71
	Bi-BM3D	**22.84**	24.50	**24.87**	**26.81**	**19.64**	**23.82**	**23.75**
100	BM3D [5]	21.91	**23.25**	23.72	25.27	18.62	22.75	22.58
	Bi-BM3D	**22.01**	23.23	**23.76**	**25.34**	**18.64**	**22.86**	**22.64**



(a1) Original image (b1) Noisy (14.16 dB) (c1) BM3D (24.99 dB) (d1) Bi-BM3D (25.08 dB)

(a2) Original image (b2) Noisy (10.63 dB) (c2) BM3D (26.79 dB) (d2) Bi-BM3D (26.81 dB)

FIGURE 4. Visual comparison of our method with the original BM3D method on the RNI6 dataset [22] under the setting of Gaussian (σ = 50, 75). Quantitative PSNR (dB) results are listed below the image.

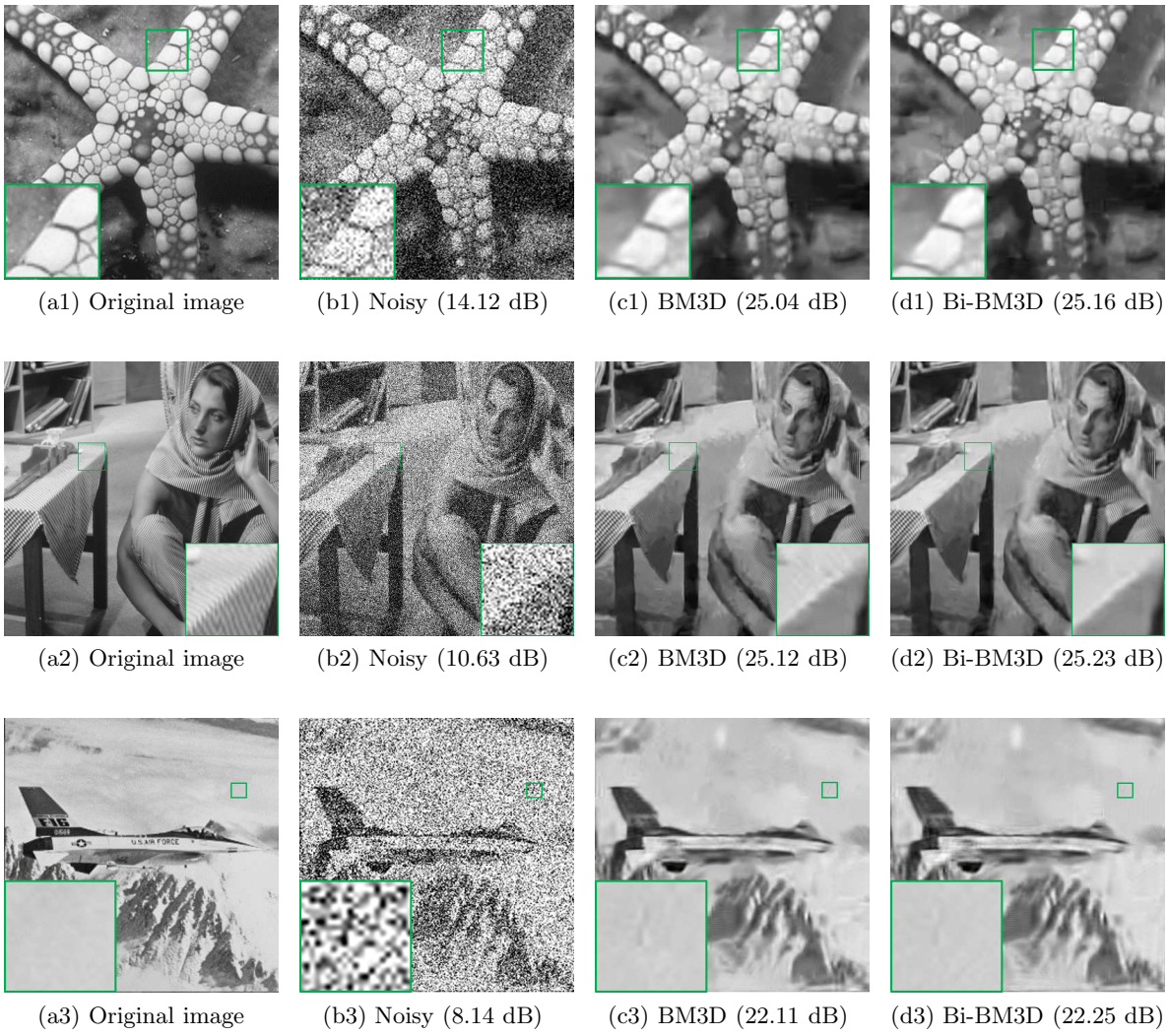
intense isolated noise altogether. In comparison, our algorithm recovers the actual texture and structures without compromising on removing noise from the images.

4.5. **Results in Set12.** We conducted comparative experiments with the original BM3D algorithm on the Set12 dataset. We generate noisy synthetic images by adding AWGN with standard deviation σ ($\sigma \in \{50, 75, 100\}$) to the Set12 clean images. From Table 5 and Figure 5, the improved algorithm we proposed has a better denoising effect than the original BM3D algorithm in the case of high noise, the intense isolated noise is also removed, and the visual effect is improved.

TABLE 5. Average PSNR (dB) results of different methods on Set12 grayscale images corrupted by AWGN noise. For PSNR (dB): larger is better; best results are shown in bold font.

σ	Image	C.man	House	Peppers	Fishstar	Monarch	Airplane	Parrot	Lena	Barbara	Boats	Man	Couple
50	BM3D [5]	26.12	29.69	26.68	25.04	25.82	25.10	**25.90**	29.05	27.23	26.78	**26.81**	26.46
	Bi-BM3D	**26.26**	**29.76**	**26.80**	**25.16**	**25.94**	**25.20**	**25.90**	**29.06**	**27.27**	**26.82**	**26.81**	**26.52**
75	BM3D [5]	24.33	27.51	24.73	23.27	23.91	23.47	24.18	27.26	25.12	25.12	25.32	24.70
	Bi-BM3D	**24.35**	**27.53**	**24.82**	**23.31**	**23.96**	**23.48**	**24.19**	**27.27**	**25.23**	**25.16**	**25.34**	**24.77**
100	BM3D [5]	23.07	25.87	**23.39**	**22.10**	22.52	22.11	**22.96**	25.95	23.62	23.97	24.22	23.51
	Bi-BM3D	**23.09**	**25.99**	**23.39**	**22.10**	**22.53**	**22.25**	**22.96**	**25.99**	**23.78**	**24.02**	**24.26**	**23.61**

4.6. **Results in BSD68.** We performed a comparison experiment with BM3D [5] on a noisy grayscale image corrupted by AWGN, where the input additive Gaussian noise is σ = 50, 75, 100. Table 6 lists the comparison of the proposed method with the original BM3D algorithm. Compared with the original BM3D algorithm, we have better denoising performance on the BSD68 dataset. As can be seen in Figure 6, we selected two images



(a1) Original image (b1) Noisy (14.12 dB) (c1) BM3D (25.04 dB) (d1) Bi-BM3D (25.16 dB)

(a2) Original image (b2) Noisy (10.63 dB) (c2) BM3D (25.12 dB) (d2) Bi-BM3D (25.23 dB)

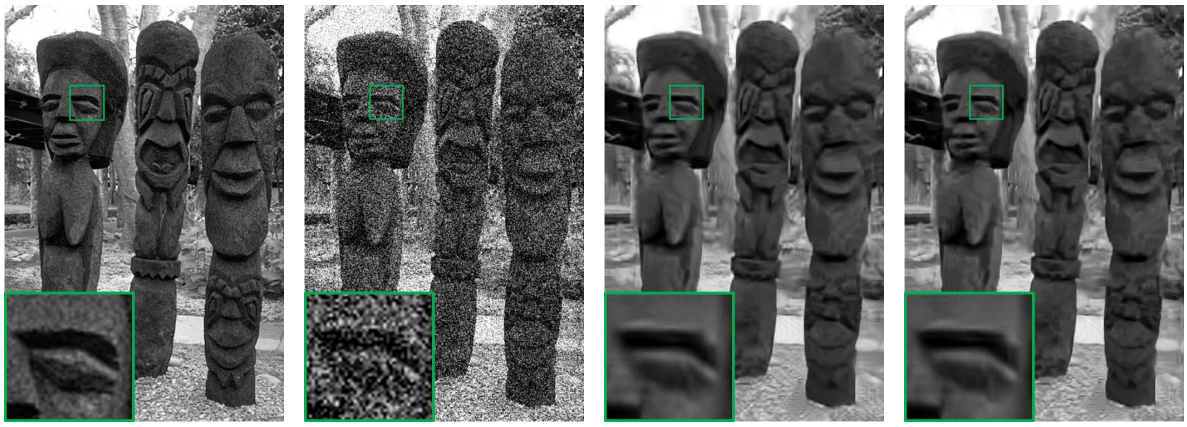
(a3) Original image (b3) Noisy (8.14 dB) (c3) BM3D (22.11 dB) (d3) Bi-BM3D (22.25 dB)

FIGURE 5. Comparison of denoised images and PSNR (dB) by BM3D and Bi-BM3D on Set12 [23]. The first row is the result of noise level $\sigma = 50$, the second row is the result of noise level $\sigma = 75$, and the third row results from noise level $\sigma = 100$.

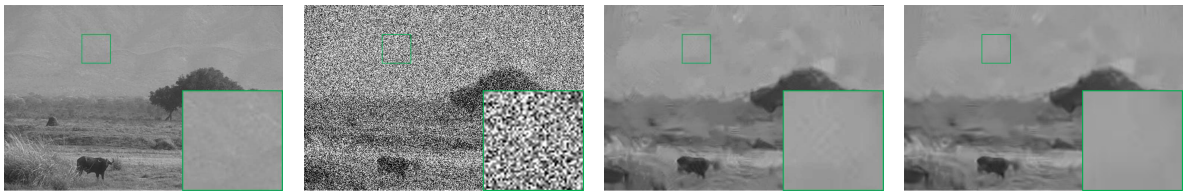
TABLE 6. Average PSNR (dB)/structural similarity index measure (SSIM) results of different methods on BSD68 [24]. For PSNR: larger is better; the best results are shown in bold font. For SSIM: larger is better; the best results are shown in bold font.

σ	50		75		100	
Metric	PSNR↑	SSIM↑	PSNR↑	SSIM	PSNR↑	SSIM
BM3D [5]	25.62	0.6926	24.21	**0.6336**	23.24	**0.5934**
Bi-BM3D	**25.65**	**0.6932**	**24.23**	0.6318	**23.28**	0.5896

of different sizes from the BSD68 dataset for denoising at noise levels of 50 and 75, respectively. However, due to the inclusion of high-frequency texture regions, Bi-BM3D has apparent advantages over original BM3D, achieving higher PSNR and SSIM results and effectively removing intense isolated noise.



(a1) Original image (b1) Noisy (14.15 dB) (c1) BM3D (22.56 dB) (d1) Bi-BM3D (22.62 dB)



(a2) Original image (b2) Noisy (10.63 dB) (c2) BM3D (25.39 dB) (d2) Bi-BM3D (25.45 dB)

FIGURE 6. Comparison of denoised images and PSNR (dB) by BM3D and Bi-BM3D on BSD68 [24]. The top is the result of noise level $\sigma = 50$; the bottom is the result of noise level $\sigma = 75$.

5. **Conclusion.** In this paper, we solve the problem that the original BM3D algorithm cannot effectively remove the intense isolated noise in high-level noise situations. We propose an improved algorithm that uses the bi-hard thresholding strategy, i.e., coefficient hard thresholding and structural hard thresholding, to replace the original BM3D hard thresholding strategy. Experiments on public datasets show that the proposed improved algorithm can effectively improve denoising performance in high-level noise situations. In the future, we will extend to RGB color image denoising and try to combine it with convolutional neural networks to improve denoising performance and visual effects.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant 62071320, the Natural Science Foundation of Shandong Province under Grant ZR2020MF038, Shandong Province Science and Technology SMEs Innovation Capacity Improvement Project under Grant 2022TSGC2195, the Science and Technology Innovation Development Project of Taian City under Grant 2020GX028 and Grant 2021GX045.

REFERENCES

[1] A. Pizurica, W. Philips, I. Lemahieu and M. Acheroy, A joint inter- and intrascale statistical model for Bayesian wavelet based image denoising, *IEEE Trans. Image Processing*, vol.11, no.5, pp.545-557, 2002.

[2] L. Sendur and I. W. Selesnick, Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency, *IEEE Trans. Signal Processing*, 2002.

[3] J. Portilla, V. Strela, M. J. Wainwright and E. P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, *IEEE Trans. Image Processing*, vol.12, no.11, pp.1338-1351, DOI: 10.1109/TIP.2003.818640, 2003.

[4] A. Buades, B. Coll and J.-M. Morel, A non-local algorithm for image denoising, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, pp.60-65, DOI: 10.1109/CVPR.2005.38, 2005.
[5] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, *IEEE Trans. Image Processing*, vol.16, no.8, pp.2080-2095, 2007.
[6] S. Gu, L. Zhang, W. Zuo and X. Feng, Weighted nuclear norm minimization with application to image denoising, *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp.2862-2869, DOI: 10.1109/CVPR.2014.366, 2014.
[7] Y. Hou, J. Xu, M. Liu, G. Liu, L. Liu, F. Zhu and L. Shao, NLH: A blind pixel-level non-local method for real-world image denoising, *IEEE Trans. Image Processing*, vol.29, pp.5121-5135, 2020.
[8] Z. Kai, W. Zuo, Y. Chen, D. Meng and Z. Lei, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, *IEEE Trans. Image Processing*, vol.26, no.7, pp.3142-3155, 2017.
[9] K. Zhang, W. Zuo and L. Zhang, FFDNet: Toward a fast and flexible solution for CNN based image denoising, *IEEE Trans. Image Processing*, 2017.
[10] S. Guo, Z. Yan, K. Zhang, W. Zuo and L. Zhang, Toward convolutional blind denoising of real photographs, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp.1712-1722, DOI: 10.1109/CVPR.2019.00181, 2019.
[11] S. Lefkimmiatis, Non-local color image denoising with convolutional neural networks, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp.5882-5891, DOI: 10.1109/CVPR.2017.623, 2017.
[12] T. Plötz and S. Roth, Neural nearest neighbors networks, *arXiv.org*, arXiv: 1810.12575, 2018.
[13] P. Liu, H. Zhang, K. Zhang, L. Lin and W. Zuo, Multi-level wavelet-CNN for image restoration, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, UT, USA, pp.886-88609, DOI: 10.1109/CVPRW.2018.00121, 2018.
[14] D. Wang, Y. Ma, Z. Pan and N. Zhang, Enhancement for low-contrast images with FitzHugh-Nagumo nonlinearity and adaptive stochastic resonance, *International Journal of Innovative Computing, Information and Control*, vol.18, no.3, pp.941-955, 2022.
[15] J. Zhang, Q. Qi, H. Zhang, Q. Du, Z. Guo and Y. Tian, Detection of bird's nest on transmission lines from aerial images based on deep learning model, *International Journal of Innovative Computing, Information and Control*, vol.18, no.6, pp.1755-1768, 2022.
[16] X. Zhang and W. Ye, An adaptive second-order partial differential equation based on TV equation and p-Laplacian equation for image denoising, *Multimedia Tools and Applications*, vol.78, no.13, pp.18095-18112, 2019.
[17] L. Ding, B. Wen, Y. Fan, C. L. Chen and T. S. Huang, Non-local recurrent network for image restoration, *arXiv.org*, arXiv: 1806.0291, 2018.
[18] D. Valsesia, G. Fracastoro and E. Magli, Image denoising with graph-convolutional neural networks, *2019 IEEE International Conference on Image Processing (ICIP)*, 2019.
[19] Y. Zhang, Y. Tian, Y. Kong, B. Zhong and Y. Fu, Residual dense network for image restoration, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.43, no.7, pp.2480-2495, DOI: 10.1109/TPAMI.2020.2968521, 2021.
[20] Y. Zhao, Z. Jiang, A. Men and G. Ju, Pyramid real image denoising network, *2019 IEEE Visual Communications and Image Processing (VCIP)*, 2020.
[21] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool and R. Timofte, SwinIR: Image restoration using Swin Transformer, *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada, pp.1833-1844, DOI: 10.1109/ICCVW54120.2021.00210, 2021.
[22] M. Lebrun, M. Colom and J. M. Morel, The noise clinic: A universal blind denoising algorithm, *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, France, pp.2674-2678, DOI: 10.1109/ICIP.2014.7025541, 2014.
[23] C. Tian, Y. Xu, W. Zuo, B. Du, C.-W. Lin and D. Zhang, Designing and training of a dual CNN for image denoising, *arXiv.org*, arXiv: 2007.03951, 2020.
[24] D. Martin, C. Fowlkes, D. Tal and J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, *Proc. of the 8th Int'l Conf. Computer Vision*, vol.2, pp.416-423, 2001.

Author Biography

Zekun Lv received his B.E. degree from the College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou, China, in 2019. He is pursuing an M.S. degree at the Guilin University of Technology, Guilin, China. His current research interests are in the areas of image processing.

Yang Geng received her B.Sc. degree from the School of Civil Engineering, the University of Jinan, in 2021. She is currently an M.Sc. student at the Qingdao Academy of Chinese Medical Sciences, Shandong University of Traditional Chinese Medicine, Jinan, China. Her current research interests include image processing, medical image processing, and analysis.

Hao Hou received his B.Sc. degree in Information and Computing Science from the School of Mathematical Science, the University of Jinan, in 2019. He is currently an M.Sc. student at the College of Intelligence and Information Engineering, Shandong University of Traditional Chinese Medicine, Jinan, China. His current research interests include image processing, medical image processing, and analysis.

Tao Lin received his B.E. degree from the School of Electronic Engineering, Chaohu University, Hefei, China, in 2021. He is pursuing an M.S. degree at the Guilin University of Technology, Guilin, China. His current research interests are in the areas of image processing.

Xiaoya Dai received her B.E. degree from the School of Computer Engineering, Jiangsu University of Technology, Changzhou, China, in 2022. She is pursuing an M.S. degree at the Guilin University of Technology, Guilin, China. Her current research interests are in the areas of image processing.

Yingkun Hou is an IEEE senior member. He received his Ph.D. degree from the School of Computer Science and Technology, Nanjing University of Science and Technology, in 2012. He is currently a professor at the School of Information Science and Technology, Taishan University, Taian, China. His current research interests include image processing, pattern recognition, and artificial intelligence.