

Research Article

Spatiotemporal Video Denoising Based on Adaptive Thresholding and Clustering

Ali Abdullah Yahya,¹ Jieqing Tan,² Benyu Su,¹ Kui Liu,¹ and Ali Naser Hadi²

¹School of Computer and Information, Anqing Normal University, Anqing 246011, China

²School of Computer and Information, Hefei University of Technology, Hefei 230009, China

Correspondence should be addressed to Ali Abdullah Yahya; aselwey1@hotmail.com

Received 24 October 2016; Revised 10 February 2017; Accepted 20 February 2017; Published 14 March 2017

Academic Editor: Allan C. Peterson

Copyright © 2017 Ali Abdullah Yahya 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.

In this paper we propose a novel video denoising method based on adaptive thresholding and K -means clustering. In the proposed method the adaptive thresholding is applied rather than the conventional hard-thresholding of the VBM3D method. The adaptive thresholding has a high ability to adapt and change according to the amount of noise. More specifically, hard-thresholding is applied on the higher noise areas while soft-thresholding is applied on the lower noise areas. Consequently, we can successfully remove the noise effectively and at the same time preserve the edges of the image, because the clustering approach saves more computation time and is more capable of finding relevant patches than the block-matching approach. So, the K -means clustering method in the final estimate in this paper is adopted instead of the block-matching method in the VBM3D method in order to restrict the search of the candidate patches within the region of the reference patch and therefore improve the grouping. Experimental results emphasize the superiority of the new method over the reference methods in terms of visual quality, Peak Signal-to-Noise Ratio (PSNR), and Image Enhancement Factor (IEF). Execution time of the proposed algorithm consumes less time in denoising than that in the VBM3D algorithm.

1. Introduction

The search for effective video denoising methods remains a major challenge for researchers. Denoising by spatial domain methods such as total variation, bilateral filter, and nonlocal mean filter is more effective for still image processing than other algorithms. However, these methods have proven their failure in preserving the image features (e.g., edges). A plethora of algorithms that are based on transform domain have been proposed to overcome the flaws of these spatial domain denoising methods [1–3]. In these algorithms, the signal is sparsely represented in the transform domain. In spite of this, for any fixed 2D transform, the great variety in natural images is unable to achieve good sparsity for all situations [4].

In recent decades, the most efficient approach in restoring video sequences takes advantage of the potential similarity between the grouped block [5]. Most algorithms in this field

have been proposed for signal processing especially for video denoising [6–11].

Lately Dabov et al. [12] have proposed a novel image denoising method based on an enhanced sparse representation in transform domain, which is known as block-matching and 3D filtering (BM3D). To the best of our knowledge, BM3D is the most efficient image denoising algorithm. In this algorithm, the initial, mutual, and similar 2D image blocks are stacked into the 3D group. Then, the group is filtered by hard-thresholding and Wiener filter, respectively. The same authors succeeded to apply the BM3D filtering scheme on video denoising, which is termed video block-matching and 3D filtering (VBM3D) [13]. In the VBM3D algorithm, the set of consecutive frames in the video sequence has been used to construct the groups. VBM3D algorithm is implemented as follows. Firstly, the groups are formed by predictive-search block-matching. Secondly, each group

is filtered by 3D transform-domain shrinkage (i.e., hard-thresholding and Wiener filtering, resp.). Finally, the last estimate of the true video is computed by aggregating all the obtained local estimates.

Despite that VBM3D method represents the state-of-the-art in video denoising, it suffers from several drawbacks where we can explain two of them as follows. One is that, in the first stage of the VBM3D filter, the hard-thresholding is incapable of distinguishing between the areas that contain more noise versus that containing less noise. As a result, the significant amount of the true signals in the less noisy areas will be removed, which will lead to deterioration of the visual quality of the output video. And then the block-matching in the VBM3D occasionally searches out of the region that contains the reference block, which will result in poor matching in the areas that heavily contaminated by noise and this would lead to blurred edges [9].

The above-mentioned disadvantages of the VBM3D filter have been extensively studied in the literature (e.g., [7, 11]). However, the challenge remains.

One of the most significant enhancements made for the VBM3D filter is the one proposed by Maggioni et al. [5], known as the video block-matching and 4D filtering (VBM4D). The mutually similar volumes in the VBM4D have been grouped rather than the blocks in the VBM3D method. In the VBM4D filter, the tracking blocks along the trajectories are used to construct the 3D spatial temporal volumes, and the mutually similar volumes have been grouped together by stacking them along the 4th D. The authors in [5] have succeeded to get better results than that of the VBM3D. However, VBM4D method suffers from high computational cost.

In order to conquer the above-stated weaknesses of the VBM3D filter, we propose to replace the steady hard-thresholding by adaptive thresholding (i.e., hard-thresholding and soft-thresholding). The hard-thresholding is applied to the areas that are heavily desecrated by noise, while the soft-thresholding is applied to the slight noise areas. Applying the hard-thresholding to the heavy noise areas will play a significant role in removing the noise effectively, while applying the soft-thresholding to the slight noise areas will assist in maintaining the edges. To avoid the second drawback of the VBM3D filter, we propose to replace the block-matching in the final estimate by the K -means clustering. Applying the K -means clustering will allow us to find the relevant patches professionally better than the block-matching, which will lead to improvement of the grouping and therefore obtain sharper edges.

The remainder of this paper is organized as follows. The proposed adaptive thresholding and clustering are described in Section 2. Experimental results are presented in Section 3. Some concluding remarks are given in Section 4.

2. Proposed Algorithm

Our proposed method is based on the following stages. In the first stage (basic estimate), we group the block-matching in the noisy video for every reference block and apply

the adaptive thresholding; whereas the hard-thresholding is applied to the areas that contain more noise and less image features, the soft-thresholding is applied to the areas that contain less noise and more image features. In the second stage, we apply the k -means clustering only to the basic estimate for finding the relevant patches and apply the Wiener filter for achieving further improvement in denoising.

In this paper, we consider $z(x, t) = y(x, t) + \eta(x, t)$, as the observed noisy video, where y is the true video signal, $\eta(\cdot) \sim \mathcal{N}(0, \sigma^2)$ is the independent and identically distributed white Gaussian noise, and $(x, t) \in X$ are the 3D spatiotemporal coordinates in the domain $X \in \mathbb{Z}^3$, where the first component $x \in \mathbb{Z}^2$ represents the spatial coordinates, while the last component ($t \in \mathbb{Z}$) represents the time index.

2.1. Grouping and 3D Transform. Suppose B_{xR} and B_x are the reference block and candidate block located at xR and x , respectively. The distance between B_{xR} and B_x [12] is

$$d(B_{xR}, B_x) = \frac{\|\gamma(\mathcal{T}_{2D}^{\text{ath}}(B_{xR})) - \gamma(\mathcal{T}_{2D}^{\text{ath}}(B_x))\|_2^2}{N^2}, \quad (1)$$

where γ is the adaptive thresholding, \mathcal{T}_{2D} is the normalized 2D linear transform, $\|\cdot\|$ denotes l^2 -norm, and N is the block size. By using d -distance (1), we can find a group containing blocks B_x all of which are similar to a reference block B_{xR} :

$$S_{xR}^{\text{ath}} = \{x \in X : d(B_{xR}, B_x) \leq \mathcal{F}_{\text{match}}^{\text{ath}}\}, \quad (2)$$

where $\mathcal{F}_{\text{match}}^{\text{ath}}$ is the maximum d -distance for two similar blocks.

In the next step, each block that has distance less than a predetermined threshold $\mathcal{F}_{\text{match}}^{\text{ath}}$ will be searched by the block-matching.

As in [13], in this paper we use the predictive-search block-matching within a search range $[-\omega, \omega]$.

Under the above-mentioned conditions the group will be formed by stacking the reference block B_{xR} and its candidate blocks. Afterwards, we apply the 3D transform to each group.

In the 3D transform, each 2D block will be transformed by 2D-transforming, while the coefficients are transformed by 1D-transforming [14]. The transform can afford to attain an extremely sparse representation of the true signal group ($M_{S_{xR}}^{\text{ath}}$), and therefore it will be easy to disentangle the noise by shrinkage.

2.2. Thresholding. In this paper, we propose a novel adaptive thresholding. This thresholding is achieved by shrinkage in a 3D transform domain.

Before talking about the proposed adaptive thresholding, let us take a brief description of hard-thresholding and soft-thresholding.

2.2.1. Hard-Thresholding. In the hard-thresholding, if the absolute value of any element is either less than or equal to the threshold, the element will be set to zero, while the

element will be retained if the absolute value is greater than the threshold.

$$X_{\text{Hard}} = \begin{cases} X & \text{if } |X| > \tau; \\ 0 & \text{if } X \leq \tau. \end{cases} \quad (3)$$

Typically, the results of the hard-thresholding are exceedingly smoothed. However, it is not enough for maintaining the edges of the image, where the denoised image usually suffers from blurring edges [10, 15].

2.2.2. Soft-Thresholding. In the soft-thresholding, if the absolute value of any element is either less than or equal to the threshold, the element will be set to zero, while the threshold will be subtracted from the element if the absolute value is greater than the threshold.

$$X_{\text{soft}} = \begin{cases} \text{sign}(X)(|X| - \tau) & \text{if } |X| > \tau; \\ 0 & \text{if } X \leq \tau. \end{cases} \quad (4)$$

The main idea of the soft-thresholding methods lies in that the coefficients have contributions from both the informative signal and noise. Consequently, the retained coefficients will be shrunk which will contribute to constricting the effects of noise.

2.2.3. Adaptive Thresholding Method. In this paper, we propose to apply adaptive thresholding (ATH) rather than the conventional hard-thresholding which was adopted by VBM3D filter. The proposed adaptive thresholding method is applied as the initial denoising for the 3D-transformed group, which is implemented as follows:

$$W(i, j) = \begin{cases} \sigma < T & \text{Soft-thresholding,} \\ \text{Otherwise} & \text{Hard-thresholding,} \end{cases} \quad (5)$$

where T is optional test threshold and σ is the noise standard deviation.

From (5) we can expect the following:

- (i) In the areas that contain more noise (i.e., $\sigma \geq T$) and less image features (such as edges), the proposed thresholding will highlight the role of the hard-thresholding which leads to reduction of the noise level significantly in these areas.
- (ii) In the areas that contain less noise (i.e., $\sigma < T$) and more image features, the proposed thresholding will highlight the role of the soft-thresholding that in turn leads to maintain the significant features of the image.

It is worth noting that the shrinkage in the VBM3D model is based on the hard-thresholding while the shrinkage in our model is based on the soft-thresholding.

2.3. Inverse Transform and Aggregation. After the noise reduction process that was implemented by the adaptive

thresholding, we apply the inverse transform to get 3D array of block-wise estimates

$$\widehat{M}_{S_{xR}}^{\text{ath}} = \mathcal{F}_{3D}^{\text{ath}^{-1}} \left(\gamma \left(\mathcal{F}_{3D}^{\text{ath}} \left(B_{S_{xR}}^{\text{ath}} \right) \right) \right), \quad (6)$$

where $\mathcal{F}_{3D}^{\text{ath}}$ is the 3D transform.

The groups $\widehat{M}_{S_{xR}}^{\text{ath}}$ (basic estimate) constitute an extremely redundant representation of the video because of the overlapping of the obtained block estimates. Consequently, there are several estimates for each pixel. For that, we aggregate these estimates in order to shape an estimate of the whole video.

The basic estimate of the true video is computed by weighted averaging of all the obtained block-wise estimates that are overlapped. The weights can be described as inversely proportional to the total sample variance of the estimate of the corresponding block-wise estimates [12].

2.4. Clustering. Block-matching is a supervised approach. The noisy video, in this approach, is processed through block-wise manner where the matching blocks are grouped for every reference block.

In the block-matching approach, it is so hard to define the threshold as to how similar to the reference block is acceptable; therefore, this approach has high computational cost.

Contrary to the block-matching approach that was adopted by the VBM3D, the clustering partitions the image into disjoint areas. As a result, we can get similar patches in an unsupervised manner [16].

The threshold in the block-matching is predetermined, while the threshold in the clustering is adaptively determined by comparing the proximities of the reference patch with different cluster centers. For that, the clustering approach has the capacity to find the relevant patches better than the block-matching approach.

In this paper, we adopt the the K -means clustering [17] in the final estimates, rather than the block-matching.

The basic idea of the K -means clustering algorithm is summarized as follows.

Step 1. Let k be the initial cluster centers m_1, m_2, \dots, m_n .

Step 2. Calculate the distance between each cluster center and each object.

Step 3. Appoint each object to the most similar cluster.

Step 4. After appointing all the objects, update the average of each cluster.

Step 5. Iterate Steps 2, 3, and 4 until the error function is converged.

In the proposed method, the K -means clustering is carried out only on the basic estimate $\widehat{M}_{S_{xR}}^{\text{ath}}$ rather than the block-matching which was conducted in both basic estimate and noisy video in [13].

Applying the block-matching on the basic estimate usually leads to blurred edges. This seems obvious once we look

at Figures 5 and 6. Thus, unlike VBM3D [13], the proposed method does not apply the block-matching method on the basic estimate, but it assigns the efficient k -means clustering method to partition the basic estimate.

The implementation of the K -means clustering on the basic estimate rather than the noisy video will lead to improvement of the grouping.

The main reason of applying the clustering to the basic estimate, instead of the input noisy video, is the flaw of accuracy in detecting edges in the noisy video. For that, the clustering can only be performed on the Wiener filtering stage where a denoised video basic estimate is obtainable.

2.5. Wiener Filtering. After the implementation of the basic shrinking (adaptive thresholding) on the transform coefficients, we implement the empirical Wiener filtering to improve the shrinkage which in turn will mitigate the noise.

The empirical Wiener shrinkage coefficients, which are computed from the energy of the 3D transform coefficient of the initial estimate group $\widehat{M}_{xR}^{\text{ath}}$, can be expressed as follows:

$$W_{xR}^{\text{wie}} = \frac{\left| \mathcal{F}_{3D}^{\text{wie}} \left(\widehat{M}_{xR}^{\text{ath}} \right) \right|^2}{\left| \mathcal{F}_{3D}^{\text{wie}} \left(\widehat{M}_{xR}^{\text{ath}} \right) \right|^2 + \sigma^2}. \quad (7)$$

The implementation results of the Wiener filtering in transform domain $\mathcal{F}_{3D}^{\text{wie}}$ are much more efficient and precise than those of the adaptive thresholding of the 3D spectrum of the noisy video.

The general final estimate $\widehat{M}_{xR}^{\text{wie}}$ is computed by the aggregation using the following weights:

$$w_{xR}^{\text{wie}} = \frac{1}{\sigma^2 \left\| W_{xR}^{\text{wie}} \right\|_2}. \quad (8)$$

3. Experimental Results

In this paper, we compare the performance of the proposed algorithm with the ones that are in [13, 18, 19] algorithms, in terms of visual quality, Peak Signal-to-Noise Ratio (PSNR), and Image Enhancement Factor (IEF).

As mentioned in Section 2, the capacity of VBM3D method is limited in highly noise-polluted video. Thus, in this section we focus on the heavily noisy video sequences in order to highlight the role of the proposed algorithm in improving the denoising performance in this case.

In this section, the video sequences, Miss America, Salesman, Tennis, and Bus, are used to verify the efficiency of the new video denoising algorithm. These video sequences have been polluted by the additive white Gaussian noise (AWGN).

To evaluate the performance of the proposed algorithm, we use the visual quality as a qualitative measurement and the PSNR and IEF as a quantitative one.

The proposed algorithm achieves higher performance on both noise removal and edges preservation as compared with the algorithms in [13, 18, 19].



FIGURE 1: Visual comparison of different algorithms for frame 5 of the Miss America sequence. From left to right and from top to bottom: original frame, noisy frame ($\sigma_n = 40$), result of [18] algorithm, result of [19] algorithm, result of [13] algorithm, and result of the proposed algorithm.

The experimental results are shown in Figures 1–4. These figures provide a visual comparison in the performance of the four algorithms applied to the test sequences: Miss America, Salesman, Tennis, and Bus. These sequences are degraded by AWGN with standard deviation $\sigma = 40$. From these figures, we can remark that the proposed method has higher ability to suppress the noise and sharpen the edges simultaneously, better than the other methods. Figures 5 and 6 offer a visual comparison of the performance of the four models.

As a qualitative measurement, the proposed model is the best in terms of edges conservation. This is clearly visible in the edges of the books of the Salesman sequence as well as in the lock of the hair of the Miss America sequence in Figures 5 and 6.

Applying the clustering approach in our proposed algorithm instead of the block-matching approach has contributed in the development of the performance of our algorithm, which in turn led to sharper edges. This is proved in Figures 5 and 6.

Figures 7–14 include the different values of PSNR and IEF of the four models with different noise standard deviations. These figures show that the proposed algorithm outperforms all the reference algorithms in terms of PSNR and IEF.

Table 1 lists the execution time of the VBM3D and the proposed algorithms for all noise levels. From this table, we can

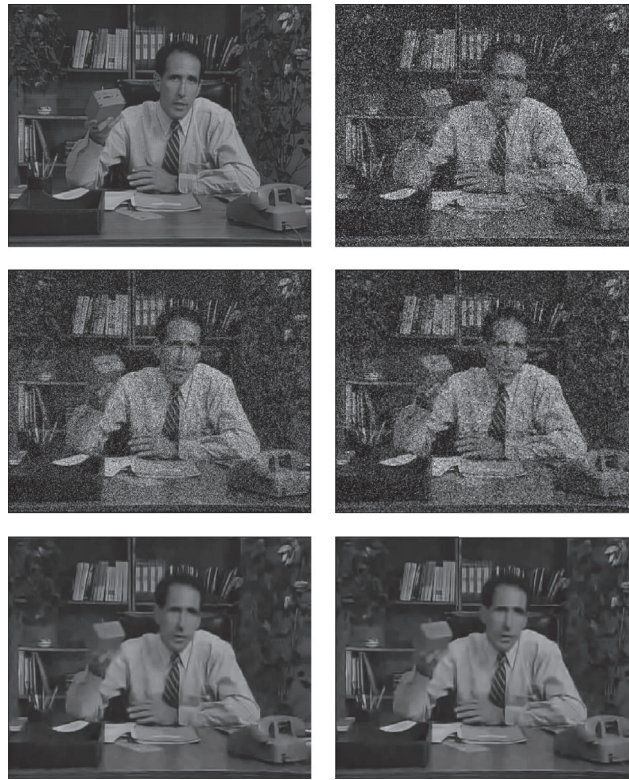


FIGURE 2: Visual comparison of different algorithms for frame 5 of Salesman sequence. From left to right and from top to bottom: original frame, noisy frame ($\sigma_n = 40$), result of [18] algorithm, result of [19] algorithm, result of [13] algorithm, and result of the proposed algorithm.

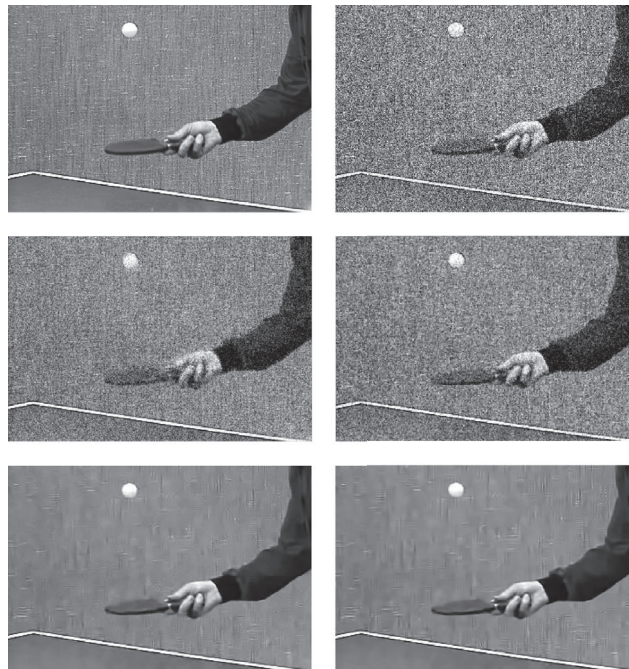


FIGURE 3: Visual comparison of different algorithms for frame 5 of Tennis sequence. From left to right and from top to bottom: original frame, noisy frame ($\sigma_n = 40$), result of [18] algorithm, result of [19] algorithm, result of [13] algorithm, and result of the proposed algorithm.



FIGURE 4: Visual comparison of different algorithms for frame 5 of Bus sequence. From left to right and from top to bottom: original frame, noisy frame ($\sigma_n = 40$), result of [18] algorithm, result of [19] algorithm, result of [13] algorithm, and result of the proposed algorithm.

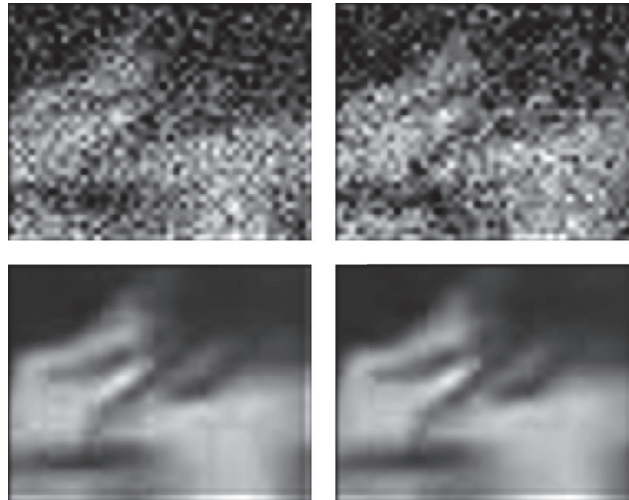


FIGURE 5: Zoom of noisy Miss America sequence ($\sigma_n = 40$). From left to right and from top to bottom: result of [18] algorithm, result of [19] algorithm, result of [13] algorithm, and result of the proposed algorithm.

TABLE 1: Computational cost of the VBM3D and the proposed algorithms for the Bus video sequence.

σ	10	20	30	40	50	60	70	80	95	100
VBM3D	0.2967 s	0.2997 s	0.2933 s	0.4944 s	0.4758 s	0.4318 s	0.4065 s	0.3960 s	0.3931 s	0.3912 s
New	0.2950 s	0.2944 s	0.2924 s	0.4863 s	0.4663 s	0.4277 s	0.4033 s	0.3948 s	0.3896 s	0.3889 s

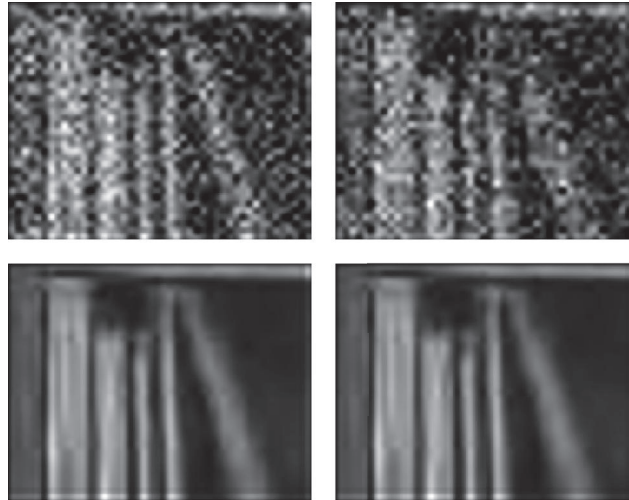


FIGURE 6: Zoom of noisy Salesman sequence ($\sigma_n = 40$). From left to right and from top to bottom: result of [18] algorithm, result of [19] algorithm, result of [13] algorithm, and result of the proposed algorithm.

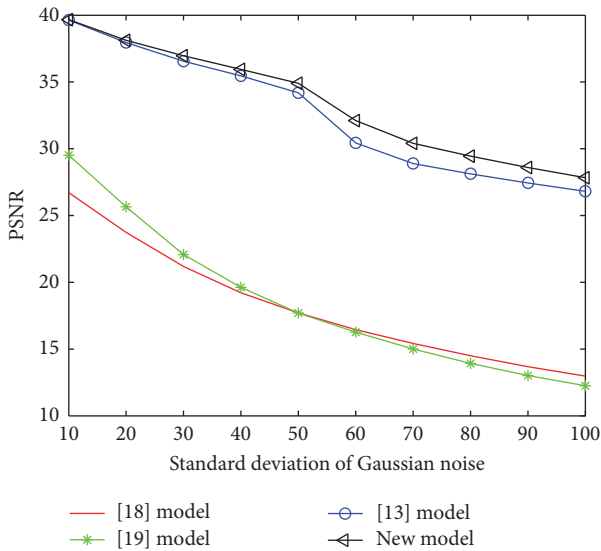


FIGURE 7: PSNR (dB) graph of [13, 18, 19] and new algorithms for various Gaussian noise for Miss America video sequence.

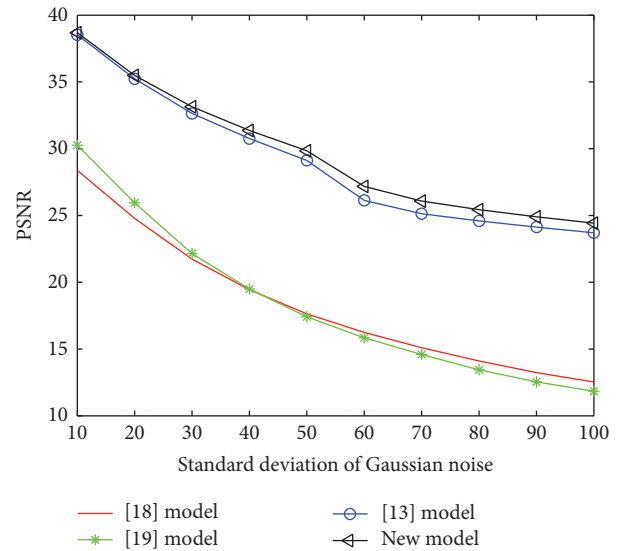


FIGURE 8: PSNR (dB) graph of [13, 18, 19] and new algorithms for various Gaussian noise for Salesman video sequence.

observe that the implementation of the proposed algorithm is faster than that of the VBM3D algorithm, which means that our proposed algorithm saves more computation time than the VBM3D algorithm. In other words, the proposed algorithm is more economical and faster.

Tables 2–9 compare the denoising performance of the four algorithms. In these tables, common video sequences have been contaminated by AWGN, with raising standard deviation $\sigma = 10$ –100. From Tables 2–5, we can see that increasing the noise standard deviation is offset by decreasing the PSNR values, which means that the denoising effect is worse. However, the PSNR values of the proposed model are the highest among the four models, which imply that the denoising effect of the new model is the best.

The data in the Tables 2–9 demonstrate that the proposed algorithm is superior to the VBM3D algorithm in all the experiments. From these tables we can also observe that our proposed algorithm has made a great improvement in all of the PSNR and IEF of the VBM3D up to 1.67 dB and 28.68, respectively.

4. Conclusion

This paper presented a novel model of video denoising based on adaptive thresholding and clustering. The proposed model has a high capacity to acclimatize and change according to the amount of the noise. Consequently, this model is able

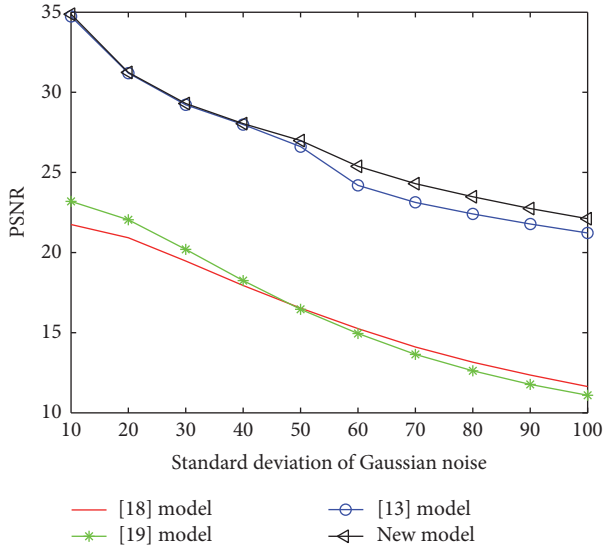


FIGURE 9: PSNR (dB) graph of [13, 18, 19] and new algorithms for various Gaussian noise for Tennis video sequence.

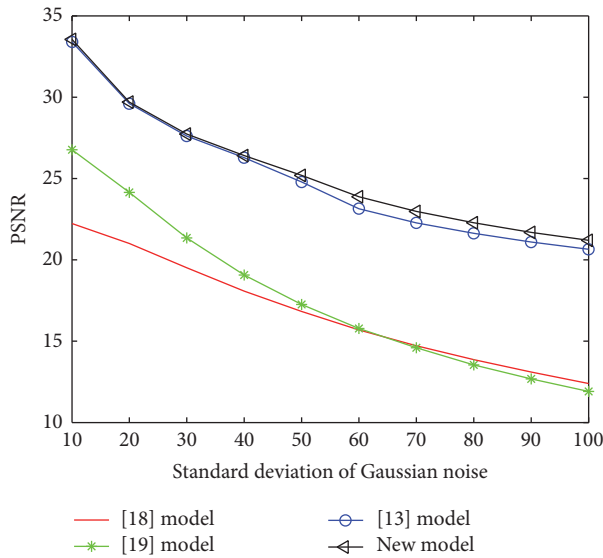


FIGURE 10: PSNR (dB) graph of [13, 18, 19] and new algorithms for various Gaussian noise for Bus video sequence.

to attenuate the noise from heavy noise video sequences effectively. In the proposed method, we applied the adaptive thresholding, instead of hard-thresholding which proved its incompetence to distinguish between the heavy noise areas and slight noise areas. Accordingly, the proposed algorithm has succeeded in maintaining the details (e.g., edges) and removing the noise strongly in comparison with the other reference methods. The applying of the clustering approach to the basic estimate in our algorithm rather than the block-matching approach in the VBM3D algorithm allowed us to get edges sharper than the ones in the VBM3D method. Numerical experiments with four different video sequences and various levels of white Gaussian noise showed that our

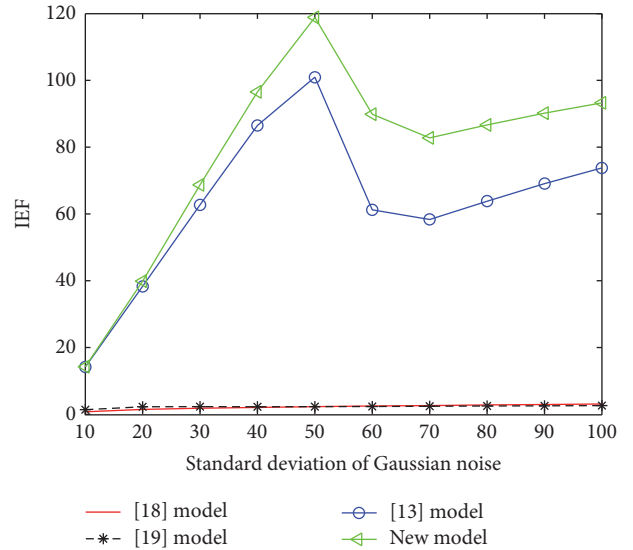


FIGURE 11: IEF graph of [13, 18, 19] and new algorithms for various Gaussian noise for Miss America video sequence.

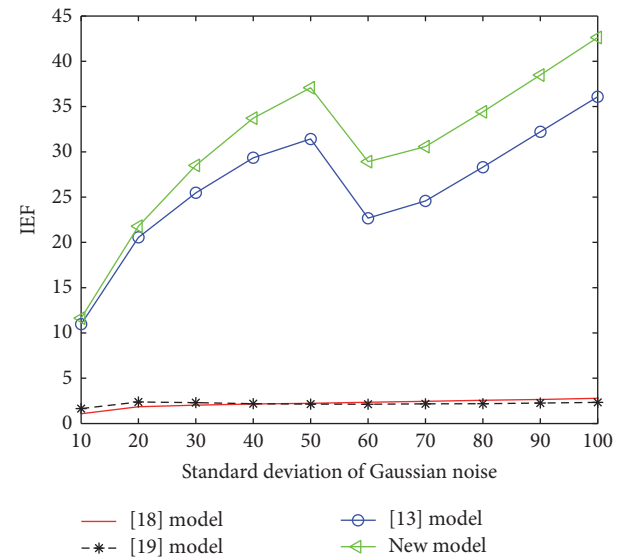


FIGURE 12: IEF graph of [13, 18, 19] and new algorithms for various Gaussian noise for Salesman video sequence.

proposed model has achieved higher noise removal gain as compared with the reference methods, in addition to 1.67 dB gain higher than that in the VBM3D algorithm. Experimental results emphasize the superiority of the proposed algorithm in terms of visual quality with obvious improvement in the PSNR and IEF. As shown in Table 1, the execution time of the proposed algorithm is less than that of the VBM3D algorithm.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

TABLE 2: PSNR of different algorithms with different Gaussian noise levels for Miss America video sequence.

σ	10	20	30	40	50	60	70	80	95	100
[18]	26.71	23.74	21.19	19.22	17.71	16.45	15.42	14.50	13.68	12.98
[19]	29.52	25.65	22.08	19.62	17.70	16.26	15.01	13.93	13.02	12.25
[13]	39.64	37.95	36.56	35.46	34.19	30.44	28.89	28.12	27.44	26.81
New	39.67	38.12	36.96	35.94	34.90	32.11	30.41	29.45	28.60	27.83

TABLE 3: PSNR of different algorithms with different Gaussian noise levels for Salesman video sequence.

σ	10	20	30	40	50	60	70	80	95	100
[18]	28.38	24.78	21.73	19.44	17.63	16.24	15.10	14.11	13.23	12.53
[19]	30.26	25.95	22.15	19.49	17.41	15.84	14.59	13.44	12.54	11.84
[13]	38.53	35.24	32.65	30.76	29.12	26.13	25.13	24.59	24.13	23.70
New	38.69	35.49	33.14	31.37	29.84	27.18	26.08	25.44	24.90	24.43

TABLE 4: PSNR of different algorithms with different Gaussian noise levels for Tennis video sequence.

σ	10	20	30	40	50	60	70	80	95	100
[18]	21.75	20.92	19.47	17.94	16.53	15.25	14.11	13.16	12.36	11.64
[19]	23.19	22.05	20.20	18.25	16.46	14.95	13.64	12.62	11.77	11.09
[13]	34.73	31.20	29.22	27.99	26.61	24.19	23.13	22.41	21.78	21.22
New	34.87	31.24	29.30	28.05	26.99	25.38	24.30	23.48	22.75	22.11

TABLE 5: PSNR of different algorithms with different Gaussian noise levels for Bus video sequence.

σ	10	20	30	40	50	60	70	80	95	100
[18]	22.23	21.01	19.51	18.08	16.83	15.69	14.72	13.87	13.10	12.40
[19]	26.77	24.15	21.35	19.07	17.26	15.78	14.60	13.54	12.68	11.91
[13]	33.40	29.61	27.61	26.28	24.79	23.14	22.27	21.63	21.10	20.65
New	33.55	29.70	27.73	26.41	25.19	23.88	22.98	22.28	21.69	21.20

TABLE 6: IEF of different algorithms with different Gaussian noise levels for Miss America video sequence.

σ	10	20	30	40	50	60	70	80	95	100
[18]	0.73	1.46	1.82	2.05	2.26	2.46	2.61	2.78	2.90	3.07
[19]	1.38	2.24	2.29	2.25	2.27	2.32	2.39	2.47	2.52	2.58
[13]	14.15	38.34	62.72	86.49	100.96	61.21	58.34	63.81	69.06	73.79
New	14.24	39.89	68.73	96.55	118.89	89.89	82.79	86.66	90.17	93.26

TABLE 7: IEF of different algorithms with different Gaussian noise levels for Salesman video sequence.

σ	10	20	30	40	50	60	70	80	95	100
[18]	1.07	1.85	2.03	2.16	2.23	2.35	2.44	2.56	2.65	2.78
[19]	1.64	2.38	2.30	2.18	2.14	2.12	2.17	2.19	2.25	2.33
[13]	10.96	20.54	25.47	29.33	31.42	22.68	24.58	28.31	32.21	36.09
New	11.63	21.79	28.51	33.71	37.07	28.91	30.57	34.42	38.48	42.62

TABLE 8: IEF of different algorithms with different Gaussian noise levels for Tennis video sequence.

σ	10	20	30	40	50	60	70	80	95	100
[18]	0.23	0.76	1.23	1.54	1.71	1.85	1.95	2.01	2.13	2.24
[19]	0.32	.099	1.46	1.64	1.71	1.72	1.74	1.80	1.87	1.98
[13]	4.57	8.12	11.55	15.51	17.63	14.53	15.48	17.15	18.78	20.39
New	4.72	8.18	11.78	15.70	19.25	19.11	20.24	21.88	23.44	24.98

TABLE 9: IEF of different algorithms with different Gaussian noise levels for Bus video sequence.

σ	10	20	30	40	50	60	70	80	95	100
[18]	0.26	0.78	1.24	1.57	1.86	2.06	2.22	2.40	2.50	2.67
[19]	0.72	1.61	1.89	1.98	2.06	2.09	2.16	2.26	2.33	2.36
[13]	3.36	5.62	7.98	10.45	11.58	11.41	12.71	14.32	16.05	17.86
New	3.48	5.74	8.21	10.76	12.70	13.54	14.97	16.63	18.38	20.26

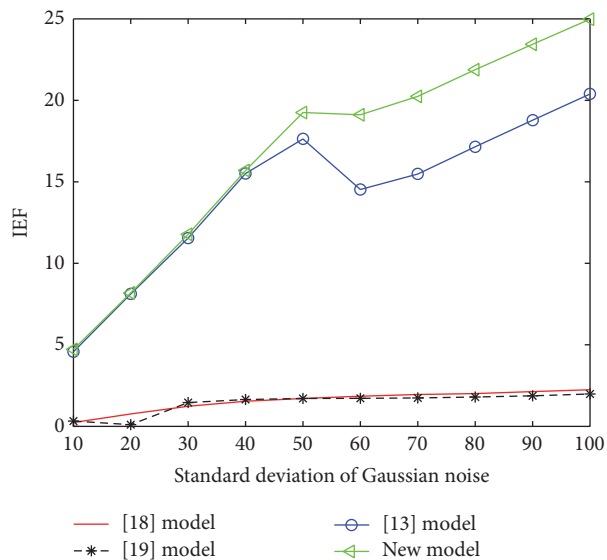


FIGURE 13: IEF graph of [13, 18, 19] and new algorithms for various Gaussian noise for Tennis video sequence.

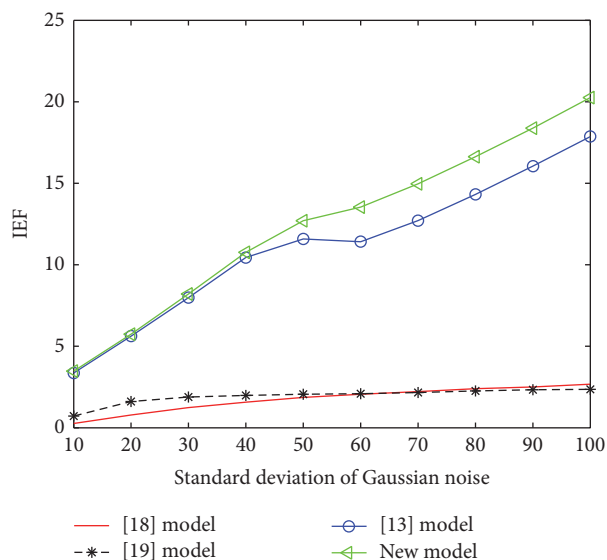


FIGURE 14: IEF graph of [13, 18, 19] and new algorithms for various Gaussian noise for Bus video sequence.

Acknowledgments

This work is supported by the Natural Science Foundation of Anhui Province under Grants nos. 1608085MF144 and 1608085QF131.

References

- [1] M. Kazubek, "Wavelet domain image denoising by thresholding and Wiener filtering," *IEEE Signal Processing Letters*, vol. 10, no. 11, pp. 324–326, 2003.
- [2] L. Tan, X. Gao, S. He, and F. Li, "Context modeling for MRI denoising in the undecimated wavelet domain," in *Proceedings of the IEEE 4th International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 3, pp. 322–326, Hainan, China, 2007.
- [3] R. Ahmed, N. Maheshwari, and P. Lalla, "Wavelet based iterative thresholding for denoising of remotely sensed optical and synthetic aperture radar images," in *Proceedings of the IEEE International Conference on Advanced Communication, Control and Computing Technologies (ICACCCT '14)*, pp. 1331–1335, May 2014.
- [4] Q. Chen and D. Wu, "Image denoising by bounded block matching and 3D filtering," *Signal Processing*, vol. 90, no. 9, pp. 2778–2783, 2010.
- [5] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3952–3966, 2012.
- [6] M. Ghoniem, Y. Chahir, and A. Elmoataz, "Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs," *Signal Processing*, vol. 90, no. 8, pp. 2445–2455, 2010.
- [7] X. Li and Y. Zheng, "Patch-based video processing: a variational Bayesian approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 1, pp. 27–40, 2009.
- [8] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, "Image and video super-resolution via spatially adaptive block-matching filtering," in *Proceedings of the International Workshop on Local and Non-local Approximation in Image Processing (LNLA '09)*, pp. 1–8, Tuusula, Finland, 2009.
- [9] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4-D nonlocal spatiotemporal transforms," in *Proceedings of the International Society for Optical Engineering (SPIE '11)*, vol. 7870, pp. 1–11, 2011.
- [10] M. Saha, M. K. Naskar, and B. N. Chatterji, "Soft, hard and block thresholding techniques for denoising of mammogram images," *IETE Journal of Research*, vol. 61, no. 2, pp. 186–191, 2015.
- [11] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, "Image and video super-resolution via spatially adaptive block-matching filtering," in *Proceedings of the International Workshop on Local and Non-Local Approximation in Image Processing*, pp. 1–8, 2008.
- [12] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [13] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *Proceedings of*

- the 15th European Signal Processing Conference (EUSIPCO '07)*, pp. 145–149, IEEE, Poznań, Poland, September 2007.
- [14] T. H. Lee and B. C. Song, “De-noising algorithm using sparse 3D transform-domain collaborative filtering and adaptive soft thresholding,” in *Proceedings of the 15th IEEE International Symposium on Consumer Electronics (ISCE '11)*, pp. 128–131, Singapore, Singapore, June 2011.
- [15] J. Ojanen and J. Heikkonen, “A soft thresholding approach for MDL denoising,” in *Proceedings of the 15th European Signal Processing Conference (EUSIPCO '07)*, pp. 1083–1087, September 2007.
- [16] L. Xu, J. Li, Y. Shu, and J. Peng, “SAR image denoising via clustering-based principal component analysis,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 6858–6869, 2014.
- [17] S. P. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [18] D. Zhang, J.-W. Han, O.-J. Kwon, H.-M. Nam, and S.-J. Ko, “A saliency based noise reduction method for digital TV,” in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '11)*, pp. 743–744, January 2011.
- [19] A. A. Yahya, J. Tan, L. Li, and M. Hu, “A novel video denoising method based on total variation and recursive temporal filtering,” *Journal of Information and Computational Science*, vol. 12, no. 13, pp. 5063–5071, 2015.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

