

# Fuzzy Logic Recursive Gaussian Denoising of Color Video Sequences via Shot Change Detection

V. Sasikala<sup>1</sup>, N. UmaMaheswari<sup>2</sup>, R. Venkatesh<sup>3</sup>

<sup>1</sup>Department of Computer Science and Engineering, University College of Engineering Ramanathapuram, Pullangudi, Tamil Nadu, India

<sup>2</sup>Department of Computer Science and Engineering, PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India

<sup>3</sup>Department of Information Technology, PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India

Email: sasivsk@yahoo.com, numamahi@gmail.com, rlvenkatesh@gmail.com

Received 28 March 2016; accepted 12 April 2016; published 15 June 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

A new technique using fuzzy in a recursive fashion is presented to deal with the Gaussian noise. In this technique, the keyframes and between frames are identified initially and the keyframe is denoised efficiently. This frame is compared with the between frames to remove noise. To do so the frames are partitioned into blocks; the motion vector is calculated; also the difference is measured using the dissimilarity function. If the blocks have no motion vectors in the block, the block of value is copied to the between frames otherwise the difference between the blocks is calculated and this value is filtered with temporal filtering. The blocks are processed in overlapping manner to avoid the blocking effect and also to reduce the additional edges created while processing. The simulation results show that the peak signal to noise ratio of the new technique is improved up to 1 dB and also the execution time is greatly reduced.

## Keywords

Keyframe Restoration, Between Frame Restoration, Motion Detection

---

## 1. Introduction

Videos and photographs play vital roles in everyday life. They are the non-destroyable and evergreen gift to remember the memories. The applications of videos and photographs are uncountable *i.e.*, they are used to control

the traffic via surveillance cameras, to suggest about a disease via video conferencing, to think about alternate to earth via satellite and also to secure our homes, offices, and streets. This may be the strong evidence to more crimes too. For example, consider that an accident takes place. A stranger who comes back can take a photograph or video of that incident. That photo contains the registration number of the vehicle, labels, the wardrobe, or sometimes the model of the vehicle etc. Unfortunately the number and label are shown but not clear. The reason is that, the stranger might have used a lower end mobile phone, or the environment will be polluted, etc. They are unwanted pixels namely noise will be added in the photograph. Hence an intelligent technique is needed to remove the unnecessary details in the photo or video moreover to get the clear and zoomed view of the number or label without spending a lot of precious time. Moreover very uncommon photos taken in some situations will get corrupted. To restore the photos, this application is very helpful.

To develop such an application, a tool with the power of converting the human thinking into mathematical model is needed. Such a tool was introduced by Lotfi Zadeh with the name of fuzzy sets in the year 1965. Generally, human thinking will be based on approximation. Such approximation can be easily represented using this fuzzy. Because whatever option we are thinking is exactly converted to rules by the developer. Over the last several years, an enormous amount of filters also filters using fuzzy tool were developed. According to the change of everyday lives, there is a gap which needs further development in all the applications. Because change is the only word that does not get changed anytime.

While looking into the previous developments, the developers really proved themselves. First, they had a close look into the pixels and processed them in a  $3 \times 3$  matrix, or  $5 \times 5$  matrix *i.e.* in a window manner. By modifying the intensity of the pixels based on averaging the neighbouring pixel values or the median value of the pixels or the center pixel values in the window, they removed noise from the photographs. They named the technique as spatial filtering [1] [2] technique. An example to such filtering technique is driven by Wiener and the name of the filter is Wiener filter. The gap *i.e.* the place which needs further development in this filter is that this filter processes not only the noisy part but also the noise free part. At the time of analyzing the output, the noise containing part is finely smoothed but the noise free portion gets blurred. This drawback is the base for further development.

While taking into account the videos, they are sequence of frames. The frame rate is different for different displays. It may be 20 fps or 30 fps according to the application *i.e.*, 20 frames or 30 frames are displayed per second. If noise is present in the video, the spatial filter should be applied to each and every frame. It will be a time-consuming process and also it produces some delay. So the developers like Yan and Yanfeng, Rakhshanfar and Amer had a close look into the details about temporal [3]-[5] data. They have tried to clear the noise by using only the temporal data. This kind of processing is named as temporal filtering. Some extent to that data, the later authors considered the data blocks too. Both the authors got their output. But the output video is in dragged *i.e.*, in slow motion effect and also it blurs the sharp edges. They are succeeded in removing the noise in videos and also reducing the checker board effect.

Both techniques contain their positive and negative sides. By integrating these two techniques namely spatio-temporal filtering [6] [7], some authors succeeded in removing the noise. But the execution time is the bottleneck to them. Because, they take all the frames to process and to produce the output.

Some authors changed the entire processing domain and initiated the concept of wavelets [8]-[16]. They moved onto frequency domain to perform the computations. They transformed the input into frequency domain via Fourier transform, cosine transform etc. Then they performed calculations and then they took the corresponding inverse transform to produce the output. They reduced the execution time but the output was less pleasant to see. To some extent to the technique, the authors like Maggioni *et al.* used the concept of data blocks. They classified the data blocks and stored the same data blocks into one volume and then performed the transform for filtering. But it required large storage space and produced considerable delay.

Again the researchers moved on to the new world of fuzzy [17] [18]. The filters were implemented using new technique which included the membership functions to categorize and the fuzzy rules to decide what actions to be done. They have concentrated on not only reducing the additive noise but also other kinds of noises (mainly impulsive noise) in color videos. It gave an added value to this technique. The added advantage of using fuzzy logic filtering technique is that the fuzzy rules can be written to maintain the precise and detailed image characteristics like edge details, color details and appearance details. Here also execution time becomes bottleneck.

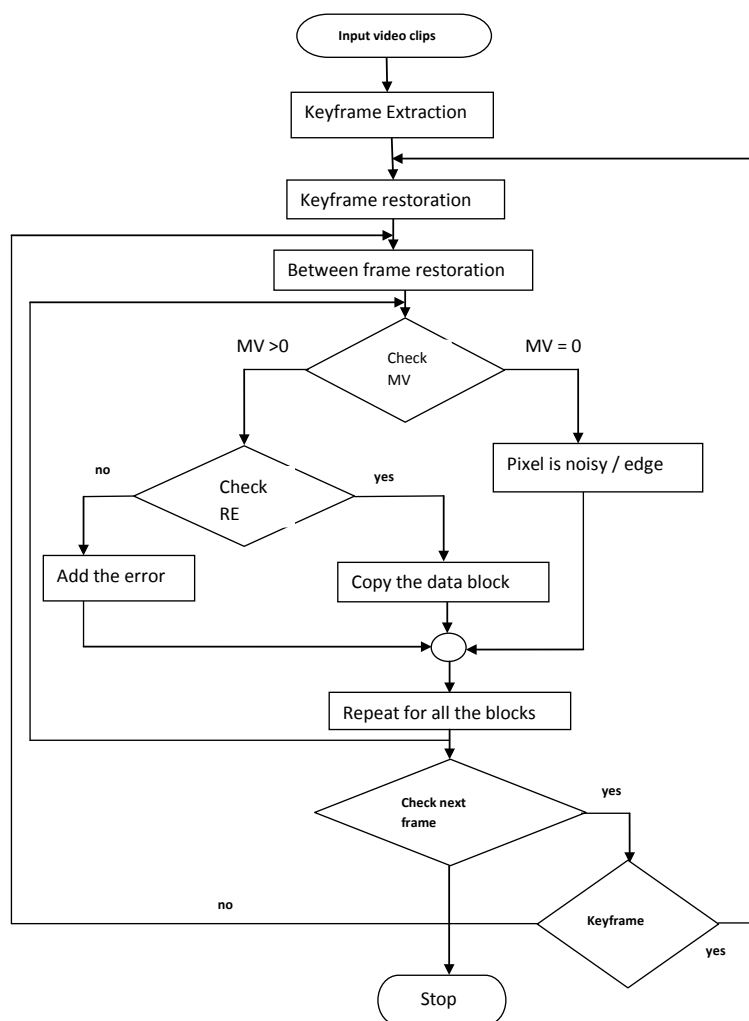
In the present work a new technique based on the above said fuzzy logic, is initialized. Here the added advantage is obtained in the output and also the execution time is reduced much better. To check the pros and cons of the algorithm, three algorithms are selected.

The first algorithm which is an extension of fuzzy directional impulsive noise removal filter is Fuzzy Directional Adaptive Recursive Temporal Filter for Gaussian denoising of color video sequences (FDARTF\_G). It is a kind of spatio temporal filter. A different approach takes place here. That is, the first frame in the video is denoised using the existing mean weighted filter. This frame is taken as the reference frame. In the subsequent frames, the difference in the movement of images between the reference frame and the next frame is calculated using angle deviation. The fuzzy rules are fired to maintain the minute details of the images that are moving. Recursively processing the subsequent  $t + 1$  frames leads to good result in time saving.

The next hybrid algorithm to be compared with the new algorithm is Color video Block Matching 3D (CBM3D) by Maggioni *et al.* This algorithm integrates the spatial and transform domain for processing. Using three steps, the denoising of video is done. In the first step, the frames are divided into blocks. The blocks contain same within the frame and in the consecutive frames are selected, tracked via motion trajectory and placed in a volume. These blocks are denoised via transform domain and the reverse is obtained to get the original output. To store and process the blocks more memory is required. If there are very large numbers of same blocks, it will slow down the execution.

The recent algorithm Fuzzy Multi channel Additive Noise Suppression (FMANS\_2) algorithm has been developed by Ponomaryov V. *et al.* This algorithm processes the video in three stages. During the initial stage, the noise pixels are identified and refined. In the second stage, two adjacent frames are processed in a manner by calculating the differences between these two frames. In the final stage, the precise details are processed individually.

The proposed model is given in **Figure 1**. This method extends the concept of FDARTF\_G. It includes the



**Figure 1.** Flow diagram of the proposed denoising algorithm.

important feature of video *i.e.*, keyframe concept. The keyframe contains the meaningful data about a particular shot of the video. Hence applying the efficient denoising algorithm on the keyframes, leads to easier denoising of subsequent frames. The technique is that the first between frame and the first denoised keyframes are taken. The motion vectors are calculated using one of the best block matching algorithm. If the motion estimation is zero, the block difference is analyzed. If it is also null, then the block is copied to the between frame else the displacement is added with the block value and then pasted. If it is equal to one, it is to be analyzed that whether the value is due to edge pixels or not. This is identified by writing a fuzzy rule using angle difference between the center pixel and the neighbouring pixel in the between frame. If it is concluded that the change is because of edge pixel, edge preservation is applied. Otherwise the displacement is again added. Thus the image details are maintained. This loop is continued until the next frame is recognized. To reduce the box effect, the blocks are overlapped.

## 2. Proposed Design

The proposed filter passes through three steps (i) Keyframe extraction (ii) Keyframe restoration and (iii) Between frame restoration to perform the filtering in a fast way. **Figure 1** gives the block diagram of the proposed fast processing filter.

### 2.1. Keyframe Extraction

To extract the keyframes, abrupt shot change detection technique is enough because the false detection won't affect the processing of denoising. The necessity of using the keyframes is that they contain more detail about the particular scene. Thus while compressing the video they are in need of large memory to store and retrieve. Nowadays there are more video coding standards available to compress the video and to retrieve the video into their original size. They give the details of number of blocks in the video, motion estimation vectors, etc., to help while playing the compressed video. These details are very useful and external motion estimation and compensation is not needed. Thus video denoising concept integrated with video coding standards reduces the bottleneck of execution time. If there is no shot change in the given video, according to the frame rate, the key frames can be set by the user. Otherwise to identify the scene change, the difference between successive frames within a block is used as a simple measure and this value is used to detect the keyframes from the given video. Initially the frames are divided into non overlapping number of blocks for the use of fast computation. The dissimilarity measure, the mean absolute value differences is calculated because of reduced computational complexity within a block is calculated. Mean Absolute Frame Differences (MAFD) between the key frame and the subsequent frames plays dual role. One is that it is used for calculating the displacement ( $d$ ) and the next one is that for key frame detection. Each block  $x \in W_n$  in the between frame  $f_B$  corresponds to a block in  $f_K$  with the displacement  $D_n$

$$f_B(x \in W_n) = f_K(x - D_n) \quad (1)$$

where  $D_n$  is called the motion vector of  $W_n$  and  $n$  is the number of blocks in the window  $W_n$ , also the mean absolute frame difference is used to calculate the minimized value of  $D_n$

$$D_n = \min \sum_{x \in W_n} |f_B(x) - f_K(x - D_n)| \quad (2)$$

$f_x(i, j)$  is the pixel intensity at position  $(i, j)$  in the current frame and  $f_{x-1}(i, j)$  is the pixel intensity in the previous frame.  $8 \times 8$  is selected as the block size.

### 2.2. Keyframe Restoration

The keyframe is processed with the existing efficient denoising algorithm. The function  $F_d(\cdot)$  is the denoising function to be applied on the keyframe  $K_1^n$  which is the first (subscript, 1) noisy (superscript,  $n$ ) keyframe. After applying the function, we get the output  $K_1^d$  of denoised (superscript,  $d$ ) version of the first (subscript, 1) keyframe. In the previous step, the frames are splitted into non overlapping blocks and block matching is applied to compute the motion compensated block. If the motion vector is zero and the block difference is negligible, then the block value will be copied to the between frame else if motion vector is zero and the block difference is

not zero, then the pixel difference is added with the keyframe block value. If motion vector is not zero it will be due to noise or edge pixels. The corresponding fuzzy rule is written as

If ((min(angle deviation between the center and diagonal pixels) > 0.1) OR (min(angle deviation between the center and adjacent pixels) > 0.1)) then noise present in the block else the edge details present in the block.

By using the above rule executing in a  $3 \times 3$  window the exact block is selected and processed accordingly.

### 2.3. Between Frame Restoration

In the between frame, decision to work on noisy pixel or edge pixel restoration is taken using the above said rule and processed accordingly. If it is noisy, then any existing temporal restoration algorithm is applied else the edge feature of the image is preserved by adjusting the pixel values related to the neighbouring pixel value. This kind of processing uses  $3 \times 3$  matrix to avoid the blocking structure and the creation of additional edges generating ghosting effect. This process is repeated while encountering a new keyframe in the video sequences.

### 2.4. Algorithm 1

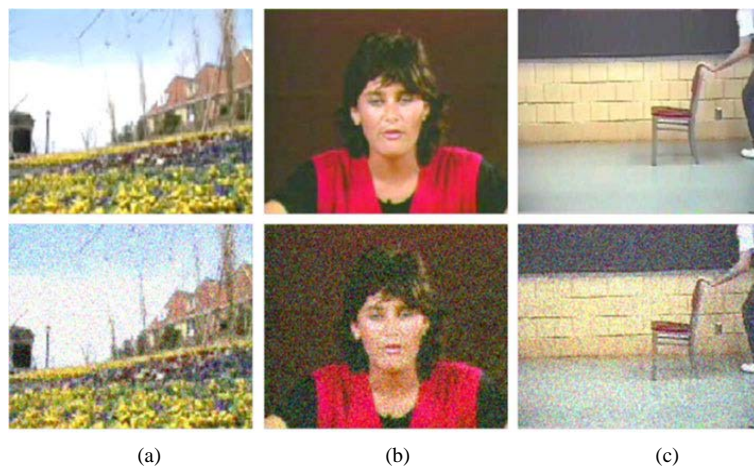
Input: RGB noisy color video  $Z$

1. Extract the keyframes from the given video
  - If  $MAFD \geq 14$ , Select the frames as keyframes  $K_t$
  - The video frames in order  $K_1, B_{11}, B_{12}, \dots, B_{1n}, K_2, B_{21}, B_{22}, \dots, B_{2n}, \dots, K_t, B_{t1}, B_{t2}, \dots, B_{tn}, K_n$
2. Denoise the keyframe using existing denoising method
3. Calculate the motion vectors and block differences
  - a. For  $mv = 0$  and no difference:
    - Copy and paste the values from the keyframe in the same location
  - b. For  $mv = 0$  but difference in blocks
    - The difference is added with the keyframe values and then pasted in the same location
  - c. For  $mv > 0$  angle difference is calculated to decide for noise pixel or edge pixel
4. Check for the next keyframe. If yes, go to step 2, else repeat step 3

Output: The denoised video sequences in the same order.

## 3. Experimental Results

This section presents the performance obtained by the proposed algorithm and it is compared with the existing algorithms (CBM3D, FDARTF\_G and FMANS\_2) which are presented earlier. The video sequences are corrupted by various levels of Gaussian noise VAR = 0.001, 0.01, 0.02, 0.03 to corrupt the input sequences. The original sequences of Flowers, Miss America, Chair and also the Gaussian noise corrupted sequences are given in **Figure 2** for easier comparison. **Figure 3** presents the denoising results of the 10<sup>th</sup> frame of Miss America



**Figure 2.** Original and corrupted images. (a) 10th flowers video sequence frame; (b) 10th Miss America video sequence frame; (c) 10th chair video sequence frame.

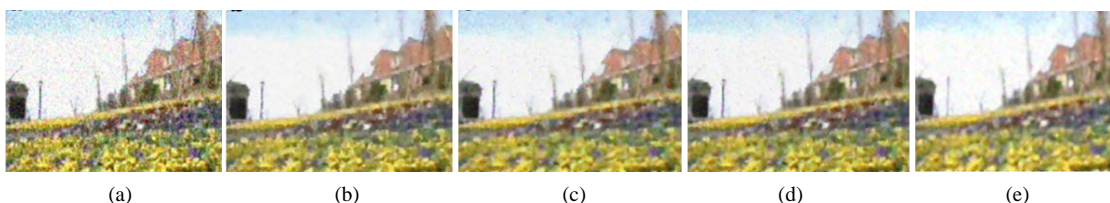
video sequence for the existing methods and for the proposed method. It demonstrates that the proposed technique performs better for all the levels of Gaussian variance. **Table 1** illustrates the results obtained by the PSNR and MAE criteria for the various noise levels and it shows that the performance the proposed method is similar to that the value obtained in FMANS\_2 and it is outperformed by FDARTF\_G. The results of the flowers video sequence are given in **Figure 4**. Based on the results, the performance driven by the proposed filtering is good for the medium noise levels of the Flowers video sequence. **Table 2** gives the PSNR and MAE values of the corresponding flower video sequence and depicts that for low noise levels FMANS\_2 works better and for high noise levels FDARTF\_G performs well. From **Figure 5** the denoising capability for the 10<sup>th</sup> frame of chair video sequences, it says that the proposed method gives the best results in preserving the image details and edges. **Figure 6** and **Figure 7** shows the denoising performance in terms of PSNR and MAE values. As a final point the proposed framework achieves the PSNR improvement up to 1 dB.

#### 4. Conclusion

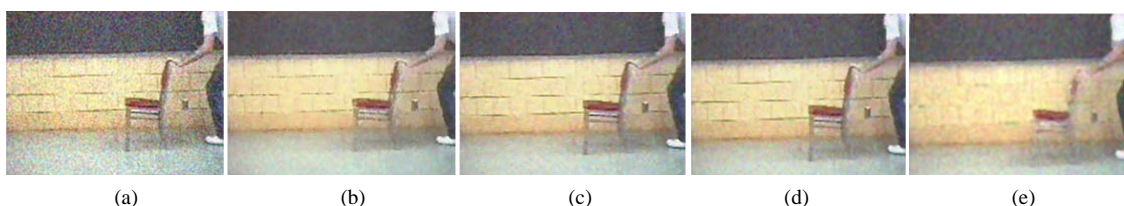
The denoising work is differently projected in this paper. Processing the keyframes and between frames recur-



**Figure 3.** The 10th frame of Miss America video sequence. (a) Frame corrupted by Gaussian noise VAR = 0.005; (b) Proposed algorithm; (c) FDARTF\_G; (d) CBM3D; (e) FMANS\_2.



**Figure 4.** The 10th frame of Flowers video sequence. (a) Frame corrupted by Gaussian noise VAR = 0.01; (b) Proposed algorithm; (c) FDARTF\_G; (d) CBM3D; (e) FMANS\_2.



**Figure 5.** The 10th frame of chair video sequence. (a) Frame corrupted by Gaussian noise VAR = 0.01; (b) Proposed algorithm; (c) FDARTF\_G; (d) CBM3D; (e) FMANS\_2.

**Table 1.** Comparative results of Miss America video sequence.

Filters Noise (VAR)	Proposed Algorithm		CBM3D		FDARTF_G		FMANS_2	
	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE
0.001	36.81	<b>4.52</b>	36.46	4.74	35.68	5.36	<b>36.83</b>	4.62
0.01	<b>33.97</b>	<b>8.43</b>	32.53	8.79	31.69	9.7	32.95	8.65
0.02	<b>30.51</b>	10.43	29.65	10.42	28.84	<b>10.31</b>	30.07	10.48
0.03	<b>27.9</b>	<b>10.91</b>	27.39	12.03	26.26	12.82	27.83	11.83

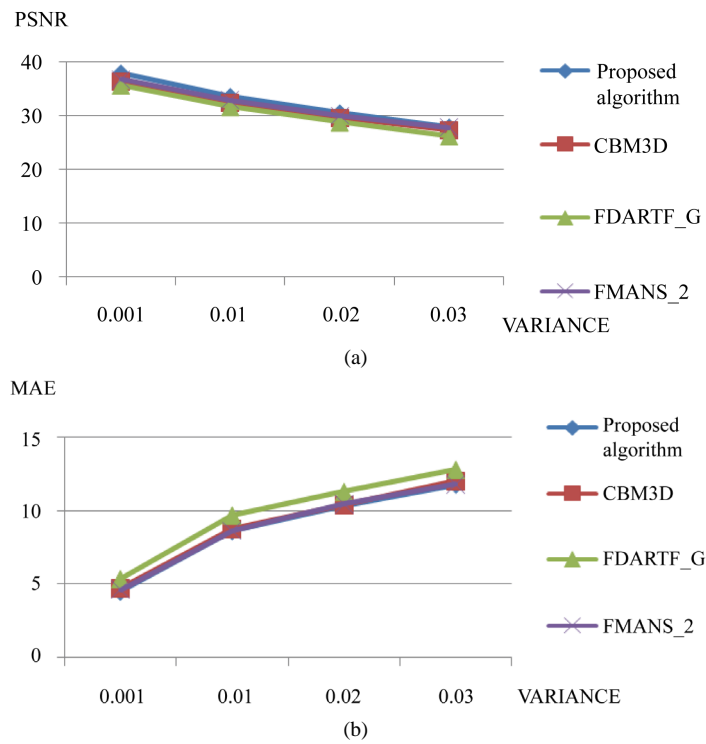


Figure 6. (a) PSNR values; (b) MAE values for the different methods on Miss America video for VAR = (0.001, 0.01, 0.02, 0.03) values.

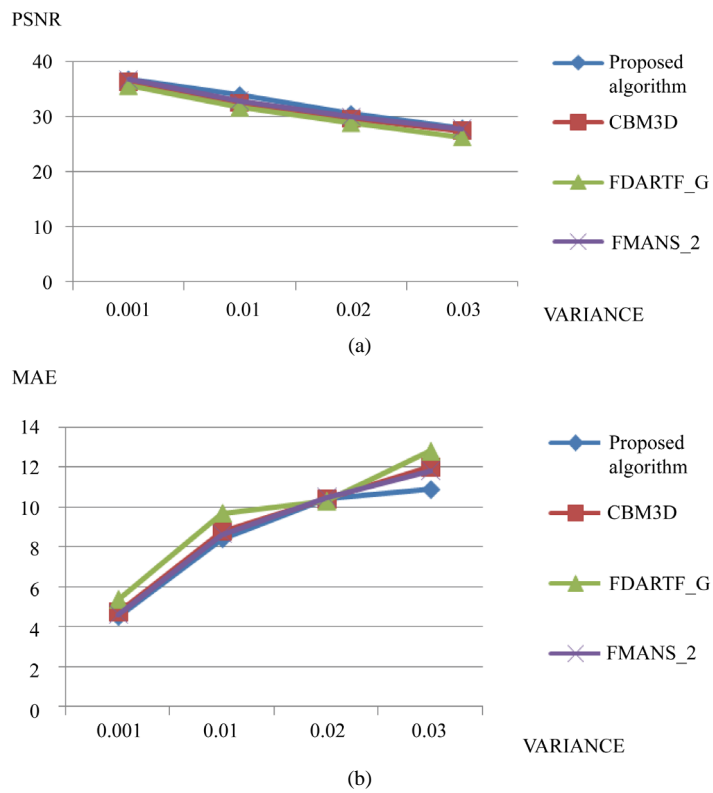


Figure 7. (a) PSNR values; (b) MAE values for the different methods on flowers video for VAR = (0.001, 0.01, 0.02, 0.03) values.

**Table 2.** Comparative results of flower video sequence.

Filters Noise (VAR)	Proposed Algorithm		CBM3D		FDARTF_G		FMANS_2	
	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE
0.001	<b>37.91</b>	<b>4.52</b>	36.44	4.72	35.66	5.34	36.80	4.60
0.01	<b>33.57</b>	<b>8.61</b>	32.51	8.76	31.67	9.60	32.93	8.63
0.02	<b>30.51</b>	<b>10.39</b>	29.63	10.40	28.82	11.30	30.05	10.46
0.03	<b>27.89</b>	<b>11.79</b>	27.37	12.01	26.24	12.80	27.81	11.81

sively throughout the video gives a pleasant vision without any delay. At any cost, only one frame is processed considering all the image features and often, the variations between blocks of two frames are calculated. If there is no motion vector and also no variations between the blocks, the values are just copied and pasted to the between frames. To effectively denoise the forthcoming frames with reference to the keyframe, this process is executed recursively till the end until a new keyframe is reached. This technique is very much useful for real-time processing and this technique can be adapted for getting the super resolution images also. Instead of executing the denoising algorithm in the key frames, any existing super resolution algorithm can be applied to get the high resolution frames. If this denoising technique is combined with the video coding standards, there will be a great reduction in the execution time. The denoising of keyframes is considered as the main part in this technique and it is very useful for the video summarization applications of various surveillance videos, sensor networks etc. This technique can also be very useful for video indexing on internet.

## References

- [1] Tomasi, C. and Manduchi, R. (1998) Bilateral Filtering for Gray and Color Images: *Proceedings of the 6<sup>th</sup> International Conference on Computer Vision*, 4-7 January 1998, 839-846. <http://dx.doi.org/10.1109/iccv.1998.710815>
- [2] Elad, M. (2002) On the Origin of the Bilateral Filter and Ways to Improve It. *IEEE Transactions on Image Processing*, **11**, 1141-1151. <http://dx.doi.org/10.1109/TIP.2002.801126>
- [3] Rajagopalan, R. and Orchard, M. (2002) Synthesizing Processed Video by Filtering Temporal Relationships. *IEEE Transactions on Image Processing*, **11**, 26-36. <http://dx.doi.org/10.1109/83.977880>
- [4] Balster, E.J. and Ewing, R.L. (2006) Combined Spatial and Temporal Domain Wavelet Shrinkage Algorithm for Video Denoising. *IEEE Transactions on Circuits and Systems for Video Technology*, **16**, 220-230. <http://dx.doi.org/10.1109/TCSVT.2005.857816>
- [5] Lee, S.-W. Maik, V., Jang, J.-H., Shin, J. and Paik, J. (2005) Noise Adaptive Spatio-Temporal Filter for Real-Time Noise Removal in Low Light Level Images. *IEEE Transactions on Consumer Electronics*, **51**, 648-653. <http://dx.doi.org/10.1109/TCE.2005.1468014>
- [6] Ercole, C., Foi, A., Katkovnik, V. and Egiazarian, K. (2005) Spatio-Temporal Pointwise Adaptive Denoising of Video: 3d Non-Parametric Regression Approach. *First Workshop on Video Processing and Quality Metrics for Consumer Electronics*, January 2005.
- [7] Varghese, G. and Wang, Z. (2010) Video Denoising Based on a Spatio-Temporal Gaussian Scale Mixture Model. *IEEE Transactions on Circuits and Systems for Video Technology*, **20**, 1032-1040. <http://dx.doi.org/10.1109/TCSVT.2010.2051366>
- [8] Rajpoot, N., Yao, Z. and Wilson, R. (2004) Adaptive Wavelet Restoration of Noisy Video Sequences. *Proceedings of the IEEE International Conference on Image Processing*, Singapore, 24-27 October 2004, 957-960. <http://dx.doi.org/10.1109/icip.2004.1419459>
- [9] Amer, A. and Schreder, H. (1996) A New Video Noise Reduction Algorithm Using Spatial Subbands. *Proceedings of the International Conference on Electronic Circuits and Systems*, Rhodes, 13-16 October 1996, 45-48. <http://dx.doi.org/10.1109/ICECS.1996.582658>
- [10] Selesnick, W. and Li, K. (2003) Video Denoising Using 2d and 3d Dual-Tree Complex Wavelet Transforms. *Proceedings of the SPIE Wavelet Applications in Signal and Image Processing*, 14 November 2003, 607-618. <http://dx.doi.org/10.1117/12.504896>
- [11] Zhang, D., Han, J.-W.O., Kwon, J., Nam, H.-M. and Ko, S.-J. (2011) A Saliency Based Noise Reduction Method for Digital TV. *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE'11)*, Las Vegas, 9-12 January 2011, 743-744. <http://dx.doi.org/10.1109/icce.2011.5722840>



- [12] Jovanov, L., Pizurica, A., Schulte, S., Schelkens, P., Munteanu, A., Kerre, E. and Philips, W. (2009) Combined Wave Let-Domain and Motion-Compensated Video Denoising Based on Video Codec Motion Estimation Methods. *IEEE Transactions on Circuits and Systems for Video Technology*, **19**, 417–421. <http://dx.doi.org/10.1109/TCSVT.2009.2013491>
- [13] Maggioni, M., Boracchi, G., Foi, A. and Egiazarian, K. (2012) Video Denoising, Deblocking, and Enhancement through Separable 4-D Nonlocal Spatiotemporal Transforms. *IEEE Transactions on Image Processing*, **21**, 3952-3966. <http://dx.doi.org/10.1109/TIP.2012.2199324>
- [14] Rusanovskyy, D. and Egiazarian, K. (2005) Video Denoising Algorithm in Sliding 3D DCT Domain. *Lecture Notes in Computer Science*, **3708**, 618-625. [http://dx.doi.org/10.1007/11558484\\_78](http://dx.doi.org/10.1007/11558484_78)
- [15] De Haan, G. (1999) IC for Motion-Compensated de Interlacing, Noise Reduction, and Picture Rate Conversion. *IEEE Transactions on Consumers Electronics*, **45**, 617-624. <http://dx.doi.org/10.1109/30.793549>
- [16] Seran, V. and Kondi, L.P. (2007) New Temporal Filtering Scheme to Reduce Delay in Wavelet-Based Video Coding. *IEEE Transactions on Image Processing*, **16**, 2927-2935. <http://dx.doi.org/10.1109/TIP.2007.909316>
- [17] Rosales-Silva, A., Ponomaryov, V. and Gallegosfunes, F. (2007) Fuzzy Directional Adaptive Recursive Temporal Filter for Denoising of Video Sequences. *Proceedings of the Artificial Intelligence 6<sup>th</sup> Mexican International Conferences on Advances in Artificial Intelligence*, 2007, 660- 670. [http://dx.doi.org/10.1007/978-3-540-76631-5\\_63](http://dx.doi.org/10.1007/978-3-540-76631-5_63)
- [18] Zlokolica, V., De Geyter, M., Schulte, S., Pizurica, A., Philips, W. and Kerre, E. (2005) Fuzzy Logic Recursive Change Detection for Tracking and Denoising of Video Sequences. *IS&T/SPIE Symposium on Electronic Imaging*, 14 March 2005. <http://dx.doi.org/10.1117/12.585854>