Available onlinewww.ejaet.com

European Journal of Advances in Engineering and Technology, 2017, 4(8): 584-591

Research Article ISSN: 2394 - 658X

An Improved VBM3D Filtering Technique for Removal Noise in Video Signals

Himanshu Makkar and Onkar Singh Lamba

Department of Electronics and Communication Engineering, Suresh Gyan Vihar University, Jaipur, India
himanshu.makkar@mygyanvihar.com

ABSTRACT

This paper presents, real-time noise removal filter from, video signals, a joint-prefiltering algorithm. A real-time video denoising filter; a great number of, digital video applications inspire the study in renovation or enhancement methods to get better the visual quality in the occurrence of noise. Video Block-Matching and 3D joint filter shortened as VBM3D, is one of the best recent video denoising filters. We speed up this filter for real-time applications by simplifying, the algorithm as well as optimizing the codes, while preserving its good denoising presentation.

Keywords: real-time filtering, pre-filtering, peak signal to noise ratio (PSNR), video block-matching and 3D collaborative filter (VBM3D), Wiener Filter

INTRODUCTION

Digital Signals, in form of video signals, are just about despoiled by noise, during acquirement footage, dealing out and broadcast also depends on the camera parameters. The noise; in video sequence not only despoiled quality but also affects the efficiency of additional processing. Therefore, noise deduction from video signals, is main because it improves the quality of perceived video sequences and enhances subsequent process in video between the original, video signals and restructured video signal.

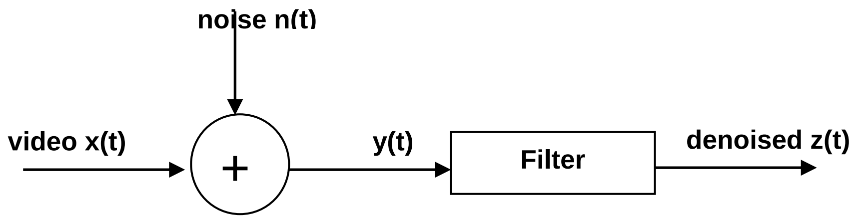

Fig. 1 Flowchart for denoising video signal [14]

Consider a original, video signal $x(t)$ ruined by noise $n(t)$ and the, noisy video can be articulated as:

$$y(t) = x(t) + n(t) \tag{1}$$

The task of noise elimination from, video signal is to filter degraded video frames $y(t)$ so as to minimize the diversity between filtered o/p $z(t)$ and original signal $x(t)$ the noise represents, refer to the Gaussian noise. In this paper contents, we will talk about the Video Block-Matching & 3D (VBM3D) filtering algorithm. In addition, we proposed a real-time implementation of simplified version of VBM3D.

GENERAL SCHEME OF VBM3D

As it was mentioned in earlier that noise filtering techniques that spatio-temporal, domain filtering, transform domain filtering and motion information can be can be used together, to improve the filtering performance, but there are also some other filtering methods, that exploit correlations using combined filtering strategies. In this paper we present VBM3D filtering method which, is one of the best noise removal filter. This technique is based on, highly sparsed signal representation in local 3D transform domain. It is an extension of BM3D filtering method for images achieve the intention of noise removal performance in terms of both PSNR and Subjective visual superiority.

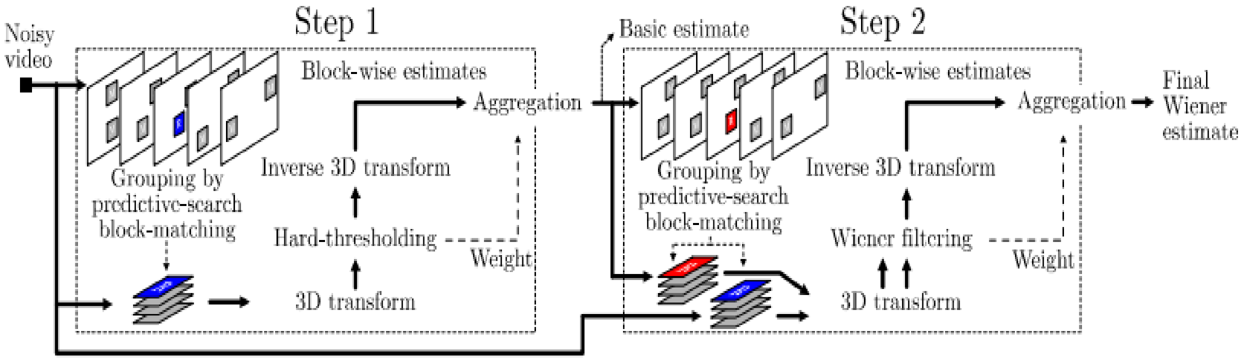

Fig. 2 Flowchart of VBM3D noise removal algorithm [17]

The general procedure consists, of the two steps. In the first place a noisy video is processed in, raster scan array and in the block wise manner. As for each reference block; a 3D array is grouped by stacking blocks, from consecutive frames, which are similar to the current processing block, a predictive search-block technique is used, for the grouping. Then a 3D transform domain reduction (first step hard-thresholding then in second place Wiener filtering) is applied, to each of the grouped 3D array since the estimate of those, obtained blocks are forever overlapped, they are aggregated by a weighted average to obtain an intermediate estimate. In second place the intermediate estimate, from the first step is used together, with a noisy video for; grouping and applying 3D collaborative, empirical filtering.

The VBM3D algorithm has three important concepts like grouping, collaborative filtering and aggregation.

Grouping

The grouping is referring to the conception of similar d-dimensional fragments, of a given signal into $d + 1$,dimensional data structure. In case of video signal the fragments can be any of the 2D blocks; and a group is a 3D array formed, by stacking together alike blocks from successive frames. Similarity between blocks is computed using the l^2 ,norm of the difference between two blocks. In order to achieve resourceful grouping, a predictive search block matching is used, to resourcefully find similar blocks. The main idea of this method, is to perform a full search in a $N_s \times N_s$ window, in current frame to get the N_B ,best matching blocks within a smaller window size, of $N_{PR} \times N_{PR}$ ($N_{PR} \ll N_S$). The window centers at the same position of the previous block. The benefit of grouping is to allow the use of high dimensional filtering, which utilizes the prospective similarity, between grouped blocks.

Collaborative filtering

Once a 3D array is obtained, from grouping, collaborative filtering can be used as second place exploit both spatial correlation inside single block and the correlation between grouped blocks. Then it is followed by reduction in the transform domain. The collaborative filtering is executed as following steps:

- execute a linear 3D transform, to the grouping
- Shrink transformed coefficients, by hard thresholding or Wiener filtering, to attenuate the noise
- Invert linear transform, to obtain estimates of grouped blocks

The advantage of collaborative filtering is to utilize both kinds of correlations, to produce a sparse demonstration of the group, and sparsity is desirable, for effective reduction in noise shrinking.

Aggregation

In general, estimates, of denoised 3D groups can be overlapped. In other terms, there can be various estimates obtained from dissimilar filtered 3D groups but have precisely the same coordinates. This leads to an over-complete demonstration of the original video, aggregation is carried out, to produce estimates of filtered 3D groups by a weighted averaging with adaptive weights.

ALGORITHM

In VBM3D filtering we believe a noisy video as:

$$z(x) = y(x) + n(x), x \in X \subset \mathbb{Z}^3 \tag{2}$$

Where y is the exact video signal, $n(\cdot) \sim (0, \sigma^2)$ is zero mean gausain noise, with variance σ^2, σ is assumed a prior known value, x is a 3D coordinate that belongs to the 3D spatio-temporial domain which refer $X \subset \mathbb{Z}^3$, and t can be expessed as:

$$x = [x_1, x_2, t]. \tag{3}$$

The first and second coordinates are the 2D spatial coordinates, in one video frame, and the third coordinate t $\subset \mathbb{Z}$, which shows the frame number. As for VBM3d algorithm. It can be divided in two steps as follow:

Step 1. *G*enerate a basic estimate, using grouping and collaborative hard-thresholding. Each reference block, Z_{xR} with $x_R \in X$ *of size* $N_{ht} \times N_{ht}$ is also taken from both the horizontal and vertical directions with a step of length of N_{step}. VBM3D groups a set of similar blocks, by using predictive search Block-Matching(PS-BM),

$$S_{xR}^{ht} = PS - BM\,(Z_{xR}) \tag{4}$$

Where Z_{xR} indicates a blocks, whose upper left curve is at xR, S_{xR}^{ht} *are similiar blocks for* Z_{xR}. All these similar blocks are grouped to form a set:

$$Z_{S_{xR}^{ht}} = \{Z_{xR} : x \in S_{xR}^{ht}\}. \tag{5}$$

Then a collaborative hard-thresholding ,is carried out within the threshold $\lambda_{3D^{\sigma}}$ to create an estimates of the set $Z_{S_{xR}^{ht}}$:

$$\hat{Y}_{S_{xR}^{ht}} = T_{3D}^{-1}\left(HARD - THR\left(T_{3D}\left(Z_{S_{xR}^{ht}}\right), \lambda_{3D^{\sigma}}\right)\right) \tag{6}$$

Where $\hat{Y}_{S_{xR}^{ht}}$ is a set with filtered blocks ans can be expressed as:

$$\hat{Y}_{S_{xR}^{ht}} = \left\{\hat{Y}_{x}^{xR} : x \in S_{xR}^{ht}\right\} \tag{7}$$

After the essential estimate $\hat{Y}^{basic}$ is calculated; by aggregation of block-wise estimates $\hat{Y}_{x}^{xR}$ according to the formula,

$$\hat{Y}^{basic} = \frac{\sum_{xR\in X}\sum_{x\in S_{xR}^{ht}} w_{xR}^{ht}\hat{Y}_{x}^{ht,xR}}{\sum_{xR\in X}\sum_{x\in S_{xR}^{ht}} w_{xR}^{ht}X_x} \tag{8}$$

Where $X_x : X \rightarrow \{0, 1\}$ is the characterstics function, of the square support, of a block situated at $x \in X$, and w_{xR}^{ht} is the influence for the current block. This weight w_{xR}^{ht} is obtained by:

$$w_{xR}^{ht} = \frac{1}{\sigma^2 N_{har}^{xR}} W_{2D} \tag{9}$$

Where N_{har}^{xR} is the quantity of non zero coefficients after hard-thresholding $T_{3D}\left(Z_{S_{xR}^{ht}}\right)$, *and* $N_{har}^{xR} > 0$ as the DC value is constantly reserved, ensuring that division by zero, not at all happens in aggregation, and W_{2D} is the 2D Kaiser window of size $N_{ht} \times N_{ht}$ which is used, for the dropping border effect.

Step 2. Obtain the absolute estimation by grouping within the essential estimate and collaborative Wiener filtering, that uses of the spectra of the related groups from the necessary estimates. For the each block $\hat{Y}_{xR}^{basic}$ *within the size of* $N_{wie} \times N_{wie}$, algorithm applies analytical search BM

$$S_{xR}^{wie} = PS - BM(\hat{Y}_{xR}^{basic}) \tag{10}$$

And based, on the set S_{xR}^{wie}, two 3D arrays are produced:

$$\hat{Y}_{S_{xR}^{wie}}^{basic} = \{\hat{Y}_{xR}^{basic} : x \in S_{xR}^{wie}\} \tag{11}$$

$$Z_{S_{xR}^{wie}} = \{Z_x : x \in S_{xR}^{wie}\} \tag{12}$$

Then the collaborative, filtering is performed in second place by, an empirical Wiener filtering and it is shown by as of,

$$\hat{Y}_{S_{xR}^{wie}} = T_{3D}^{-1}(T_{3D}\left(Z_{S_{xR}^{wie}}\right)\frac{\left(T_{3D}\left(\hat{Y}_{xR}^{basic}\right)\right)^2}{\left(T_{3D}\left(\hat{Y}_{xR}^{basic}\right)\right)^2 + \sigma^2} \tag{13}$$

The absolute estimate ($\hat{Y}^{final}$) is formed by aggregation of those of overlapped estimates is given by:

$$\hat{Y}^{final} = \frac{\sum_{xR\in X}\sum_{x\in S_{xR}^{wie}} w_{xR}^{wie}\hat{Y}_{x}^{we,xR}}{\sum_{xR\in X}\sum_{x\in S_{xR}^{wie}} w_{xR}^{wie}X_x} \tag{14}$$

With the weight of

$$w_{xR}^{wie} = \sigma^{-2}\left\|\frac{\left(T_{3D}\left(\hat{Y}_{xR}^{basic}\right)\right)^2}{\left(T_{3D}\left(\hat{Y}_{xR}^{basic}\right)\right)^2 + \sigma^2}\right\|_2^{-2} W_{2D} \tag{15}$$

Where $\|\cdot\|_2$ *denotes* l^2 standard, and W_{2D} is 2D Kaiser window of size $N_{wie} \times N_{wie}$.

COMPLEXITY ANALYSIS

In this study, complication is calculated based on the number of essential arithmetic operations, however additional factors, such as memory utilization and the number of the memory access have not been measured. The complication of VBM3D (C_{VBM3D}) and the metaphors of the parameters in the subsequent equations are exposed in table [1]

$$C_{VBM3D} = C_{VBM3D}^{ht} + C_{VBM3D}^{wiener} \tag{16}$$

Hard-thresholding stage, for the each processed block, at a large amount of M similar blocks are the extracted within the search-window of the size $N_s \times N_s$ and they stacked together as a cluster then a 3D transform and hard-thresholding are of the applied to the 3D group. In conclusion the essential estimate is obtained by aggregating of the inversed coefficients. Thus the complication of hard-thresholding stage can be articulated as:

$$C_{VBM3D}^{ht} = T\frac{n}{N_{step}^2}\left(\left(N_s^2 + zN_B N_{PR}^2\right)3N^2 + 2\left(2MC_{(N,N,N)} + C_{M,M,N^2}\right) + MN^2\right) \quad (17)$$

Wiener-filtering stage, the most processes are of the same as, those in the hard- thresholding stage, but two groups in its place of one require to be transformed. Element-wise multiplications are applied for the obtaining coefficients' reduction, which is involves a sets of weights in calculation and requires 6 arithmetic per pixel:

$$C_{VBM3D}^{wiener} = T\frac{n}{N_{step}^2}\left(\left(N_s^2 + zN_B N_{PR}^2\right)3N^2 + 4\left(2MC_{(N,N,N)} + C_{M,M,N^2}\right) + 6MN^2 + MN^2\right) \quad (18)$$

Table -1 Parameters Concerned in the VBM3D Complication Analysis

Parameter	Description
T	Total number of frames in the video signal
N	Numbers of the pixel per frame
N	2D block length
Z	Temporal search window length in group
N_S	Spatial search window length
N_{step}	Sliding pace to process every next reference block
M	Total number of the blocks in grouped 3D array
$C_{(a,b,c)}$	Numeric process is required by a multiplication of between two matrices of size $a \times b$ *and* $b \times c$

PRACTICAL RESULTS

In this section, we present and discuss some experimental results VBM3D technique and the results obtained of VBM3D filtering on two standard ballrooms (352 x 288) and vassar (640 x 480) degraded by the Gaussian noise with the variance of 20^2 are revealed in table [2] comparisons of subjective visual quality between the original and noisy and the denoised frames are illustrated in figure [3].

On one hand, the result reflect that the VBM3D filter achieves the state of the art of denoising performance in the terms of both PSNR and the subjective video superiority. On other hand due to the high complication of the algorithm, the pace at which present execution of VBM3D executes make it the hard to meet the real time necessities. We define the real time requirements as of the filter has at least 25 fps for the processing frames with the resolution of 640 x 480 under the computer platform with Intel Core i5 2.3GHz and 4GB of RAM. However, the speed of the present execution is only 1.01 fps for the vassar 640 x 480. To solve this problem, we simplify the VBM3D algorithm and also propose a fast integer execution.

Table -2 Performance of the VBM3D along with different test of the video sequences corrupted by the Gaussian noise $\sigma = 20$ in the computer Intel Core i5 2.3GHz and 4GB of RAM

$\sigma = 20$	Foreman	vassar
Resolution	352 x 288	680 x 480
Noisy (dB)	22.10	22.10
Denoised (dB)	34.42	36.25
Speed	3.46	1.01

(a)

(b)

(c) (d)

(e) (f)

Fig. 3 Examples of VBM3D filtering: The two test videos vassar and ballroom are already being degraded by Gaussian noise signal with $\sigma = 20$, and (a),(c),(e) in that order are original, noisy and the denoised frame for vassar and (b),(d),(f), respectively are original, noisy and the denoised frames for the ballroom

REAL-TIME IMPLEMENTATION OF VBM3D

In this optimizing task two approaches are used -

- Simplify VBM3Dfilter by by means of only the most in high-ranking parts for noise attenuation.
- Propose a fast integer execution of the simplified VBM3D.

In the very first approach, we want to find which parts of the VBM3D are most prominent for noise reduction. The VBM3D filter can be further divided into two steps, and each of them step has numerous sub-steps as offered previously. Some experiments are carried out to find the noise reduction ability of each sub-step. We \turn off in one sub-step and \turn on" in all the other parts of the VBM3D filter, then we record the number of filter performance. After experiments, we get that temporal correlation also contributes more than the spatial correlation in noise reduction. As a result, we choose only to use of the temporal search, the temporal transform and the hard-thresholding in the first step, but to remove the Wiener filtering part due to its high complication. At the same time, we are chainging the values of two parameters in the proposed filter settings. One of them is the number of the temporal searching frames, and reducing the range from 9 to 5. In other words, it only finds the current frame, of the two previous and the two following frames. The other is N2 (maximum length of the 3-dimension transform), using 4 in its place of 8. By doing this, the computational complication of VBM3D is significantly decreased. The comparison of the standard VBM3D and the simplified VBM3D algorithm is shown in the Table -3.

Sequences of the two video vassar (640 x 480) and ballroom (640 x 480) are used in our experiments, and the both of these two videos are despoiled by Gaussian noise with the different variances. The assessment of performance between the standard VBM3D and the simplified VBM3D is shown in Table 4. As we can see from Table 4, for the small sigma value, such as 5, even though performance of the simplified VBM3D is not as good as of the standard VBM3D caused by simplification of the algorithm, and the simplified VBM3D still has one of the good denoising capacity. This is because human eyes generally cannot tell the diversity among images which have PSNR values above 37dB. As the increments of the sigma values, the simplified VBM3D performs worse than the standard VBM3D. But in common the simplified VBM3D improves the PSNR values of the noisy video signals by 4 to 6 dB.

Moreover, it is significant to note, that the rate of the simplified VBM3D is about the 7 times faster than the speed of the standard VBM3D. However, it is still not the fast enough for the real-time applications since it needs to have at least the 25 fps. Therefore, we continue to speed up the simplified VBM3D by using second approach.

In this second approach, an integer implementation of simplified VBM3D is proposed in this approach. The algorithm comparison of proposed implementation and of the simplified VBM3D is shown in Table -5. The proposed implementation has the numerous improvements compared to the simplified VBM3D, and they are shown below.

1. In place of float type, integer type is used for all the variables.
2. Instead of the buffer whole video, proposed implementation only buffers for the 4 frames.
3. Instead of full search algo, we propose to use the modified method of the diamond search and the algorithm is described in the detail in next section.
4. Reduce the number of temporal-searching frames from 5 to 4, so all the blocks are in grouped from searched frames are utilized in Haar transform, for reducing the computational complexity.

Table -3 Comparison of the Standard VBM3D and the Simplified VBM3D Algorithm

Filters		Standard VBM3D	Simplified VBM3D
Step - 1	Spatial Search	+	-
	Spatial Transform	+	-
	Temporal Search	+	+
	Temporal Transform	+	+
	Hard Thresholding	+	+
Step - 2	Spatial Search	+	-
	Spatial Transform	+	-
	Temporal Search	+	-
	Temporal Transform	+	-
	Wiener Filtering	+	-
Temporal Searching Frames		9	5
N2(Maximum length of the haar transform)		8	4

Table -4 Comparison of performance between the standard VBM3D and the simplified VBM3D for the denoising video sequences of vassar and ballroom which are the degraded by Gaussian noise with the different variances, in the computer platform with Intel Core i5 2.3GHz and 4 GB of RAM

PSNR	Resolution 640x480Number of frames 250		Standard VBM3D	Simplified VBM3D
5/34.13	vassar	Denoised (dB)	40.74	38.38
		Speed (fps)	1.09	7.22
	ballroom	Denoised (dB)	41.44	37.74
		Speed (fps)	1.11	7.29
10/28.12	vassar	Denoised (dB)	38.21	33.67
		Speed (fps)	1.14	7.67
	ballroom	Denoised (dB)	38.69	33.15
		Speed (fps)	1.09	7.22
15/24.63	vassar	Denoised (dB)	36.57	30.23
		Speed (fps)	1.10	7.28
	ballroom	Denoised (dB)	36.71	29.97
		Speed (fps)	1.13	7.12
20/22.18	vassar	Denoised (dB)	35.32	27.85
		Speed (fps)	1.13	7.28
	ballroom	Denoised (dB)	35.20	27.24
		Speed (fps)	1.15	7.01

Table -5 Algorithm Comparison for the Proposed Implementation and a Simplified VBM3D

	Proposed Implementation	Simplified VBM3D
Data Type	integer	Float
Memory	Buffer only 4 frames	Buffer whole video
Block Matching	Modified diamond search	Full search
Temporal search window	4	5

MODIFIED DIAMOND SEARCH ALGORITHM

Step -1

Centre a large diamond search pattern (LDSP) is at a predefined search window, and search the 5 check blocks in a pre-defined order: centre, horizontal and the vertical. The first check block, with the sum squared difference has less than threshold is the final solution. If in the sum squared differences of all the check blocks are greater than the threshold, and of the minimum block distortion point, and the abbreviated as MBD, is found to be at in the centre, jump to Step 3; or else, go to Step 2.

Step -2

Construct a LDSP centred on the position of MBD point from the previous search. Search within the 5 check blocks with a pre-defined order: centre, horizontal and vertical. This first check block, with the sum squared difference has less than the threshold, in the final solution. If the sum squared differences of all the check blocks are greater than threshold, and at the minimum block distortion point, skip to Step 3; or else, repeat this step.

Step -3

Switch this search pattern from the large pattern to the small diamond search pattern (SDSP) and create a SDSP at the position of MBD point from the previous search. At the minimum block distortion among check the blocks is the final solution

Table -6 illustrates the performance of comparisons of the proposed implementation and of the simplified VBM3D. From the results, we got that the proposed implementation is of about the 4 to 5 time faster than the simplified VBM3D version. The proposed filter has above 30 fps, which meets the necessities for real-time video denoising applications. also, for video vassar which has the static background, the pro-posed implementation outperforms in the simplified VBM3D in terms of the PSNR values, and with the PSNR improvement up to the 0.7 dB. This is mainly as a result of the motion search method which used in our algorithm. The modified diamond search algorithm gives us a strong preference to the position of the orientation block, which produces more than precise calculation for the static background. As a result of, our proposed implementation is the much faster than the simplified VBM3D, and the outperforms is the simplified VBM3D for videos with the static background, just as in the video conference applications.

Table -6 Performance comparison between the standard VBM3D, and the simplified VBM3D and the proposed implementation for the denoising video sequences of the vassar and the ballroom which are being corrupted by Gaussian noise with the different variances, in the computer platform with Intel Core i5 2.3GHz and 4GB of RAM

PSNR	Resolution 640x480 Number of frames 250		Standard VBM3D	Simplified VBM3D	Proposed implementation
5/34.13	vassar	Denoised (dB)	40.74	38.38	38.41
		Speed (fps)	1.09	7.22	34.11
	ballroom	Denoised (dB)	41.44	37.74	37.73
		Speed (fps)	1.11	7.29	34.40
10/28.12	vassar	Denoised (dB)	38.21	33.67	33.92
		Speed (fps)	1.14	7.67	34.47
	ballroom	Denoised (dB)	38.69	33.15	32.78
		Speed (fps)	1.09	7.22	30.49
15/24.63	vassar	Denoised (dB)	36.57	30.23	30.94
		Speed (fps)	1.10	7.28	34.25
	ballroom	Denoised (dB)	36.71	29.97	29.82
		Speed (fps)	1.13	7.12	34.94
20/22.18	vassar	Denoised (dB)	35.32	27.85	28.37
		Speed (fps)	1.13	7.28	33.94
	ballroom	Denoised (dB)	35.20	27.24	27.80
		Speed (fps)	1.15	7.01	34.54

CONCLUSION

In this part, we have a general form of review of the video denoising algorithms and VBM3D. VBM3D has an excellent filtering ability, but the current implementation does not suit for the real-time implementations. In order to accelerate the VBM3D and conserve good filtering performance, and we simplify VBM3D algorithm and implement it in the real-time. From our experiments, we have concluded that even though that the proposed implementation has some PSNR degradation as compared with the standard VBM3D, it still has the good denoising performance, with PSNR perfection of around 4 dB over the noisy videos. Moreover, it is very important to note that the proposed new implementation is the 30 times faster than the previous standard VBM3D, and it can be used in the real-time video applications.

REFERENCES

[1] MR Amerand S Todorovic S, Sum-Product Networks for Modeling Activities with Stochastic Structure, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2012**, 1314–1321.

[2] J Bayer, C Osendorfer, D Korhammer, N Chen, S Urban and PvanderSmagt, On Fast Dropout and Its Applicability to Recurrent Networks, *International Conference on Learning Representations (ICLR)*, **2014**.

[3] W Brendel and S Todorovic, Learning Spatiotemporal Graphs of Human Activities, *IEEE Conference on Computer Vision (ICCV)*, **2011**, 778–785.

[4] Himanshu Makkar, Image Analysis Using Improved Otsu's Thresholding Method, *International Journal of Recent and Innovation Trends in Computing and Communication*, **2014**, 2(8), 2122-2126.

[5] JM Chaquet, EJ Carmona and A Fernndez-Caballero, A Survey of Video Datasets for Human Action and Activity Recognition, *Computer Vision and Image Understanding*, **2013**,117(6), 633–659.
[6] Z Cheng, L Qin, Q Huang, S Jiang, S Yan and Q Tian, Human Group Activity Analysis with Fusion of Motion and Appearance Information, *ACM International Conference on Multimedia (ACM MM)*, 2011, 1401– 1404.
[7] KM Chung, WC Kao, CL Sun, LL Wang and CJ Lin, Radius Margin Bounds for Support Vector Machines with The RBF Kernel, *Neural Computation*, **2003**,15(11), 2643–2681.
[8] H Do and A Kalousis, Convex Formulations of Radius-Margin Based Support Vector Machines, *International Conference on Machine Learning (ICML)*, **2013**.
[9] H Do H, A Kalousis and MHilario, Feature Weighting Using Margin and Radius Based Error Bound Optimization in SVMS, *European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, **2009**, 315–329.
[10] J Donahue, LA Hendricks, S Guadarrama, M Rohrbach, S Venugopalan, KSaenko and T Darrell, Long-term Recurrent Convolutional Networks for Visual Recognition and Description, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2015**,
[11] PF Felzenszwalb, RB Girshick, D McAllesterand D Ramanan, Object Detection with Discriminatively Trained Part Based Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **2010**, 32(9), 1627–1645.
[12] R Girshick, J Donahue, T Darrell and J Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2014**, 580– 587.
[13] R Gupta R, AYS Chia and D Rajan, Human Activities Recognition using Depth Images, *ACM International Conference on Multimedia (ACM MM)*, **2013**, 283–292.
[14] A Karpathy, G Toderici, S Shetty, T Leung, R Sukthankar and LFei-Fei, Large-Scale Video Classification with Convolutional Neural Networks, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2014**.
[15] Oge Marques, *Practical Image and Video Processing Using MATLAB*, John Wiley & Sons, Inc. publication, **2011**.
[16] John W Woods, *Multidimensional Signal, Image, and Video Processing and Coding*, Second Edition, Academic Press is an Imprint of Elsevier, Inc. Publication, **2012**.
[17] K Dabov, A Foi and K Egiazarian, Video Denoising by Sparse 3D Transform Domain Collaborative Filtering, *Proceedings of 15th European Signal Processing Conference*, EUSIPCO, Poland, **2007**.