Journal of Computer & Robotics 14 (1), 2021 21-32

Computer & Robotics

An Improved Real-Time Noise Removal Method in Video Stream based on Pipe-and-Filter Architecture

Vahid Fazel Asl[a], Babak Karasfi[a,*], Behrooz Masoumi[a], Mohammad Reza Keyvanpour[b]

[a] *Faculty of Computer and Information Technology Engineering, QIAU, Qazvin, Iran.*
[b] *Computer Engineering Department, Alzahra University, Tehran, Iran.*

Received 3 April 2021; accepted 27 June 2021

Abstract

Automated analysis of video scenes requires the separation of moving objects from the background environment, which could not separate moving items from the background in the presence of noise. This paper presents a method to solve this challenge; this method uses the Directshow framework based on the pipe-and-filter architecture. This framework trace in three ways. In the first step, the values of the MSE, SNR, and PSNR criteria calculate. In this step, the results of the error criteria are compared with applying salt and pepper and Gaussian noise to images and then applying median, Gaussian, and Directshow filters. In the second step, the processing time for each method check in case of using median, Gaussian, and Directshow filter, and it will result that the used method in the article has high performance for real-time computing. In the third step, error criteria of foreground image check in the presence or absence of the Directshow filter. In the pipe-and-filter architecture, because filters can work asynchronously; as a result, it can boost the frame rate process, and the Directshow framework based on the pipe-and-filter architecture will remove the existing noise in the video at high speed. The results show that the used method is far superior to existing methods, and the calculated values for the MSE error criteria and the processing time decrease significantly. Using the Directshow, there are high values for the SNR and PSNR criteria, which indicate high-quality image restoration. By removing noise in the images, you could also separate moving objects from the background appropriately.

Keywords: Image processing; background removal; Directshow framework; pipe-and-filter architecture

1.Introduction

Images and video sequences are often affected by noise due to inappropriate acquisition, transmission, or recording. In general, video data tend to be noisier than a single image due to the high-speed capturing rate of the video camera[1]. Video noise removal is necessary for video application systems, such as intelligent video surveillance[2] and traffic observations.

The main aims of these video application systems are to provide an automatic interpretation of scenes and analyze the actions and interactions of the observed items based on the information acquired by cameras[3]. Cameras generate existing noise in images. The most current noises in video application systems do not make moving objects separated from the background environment appropriately[4]. In these systems, obtaining foreground regions is one of the most critical requirements. Background subtraction techniques[5]are the most popular choice to remove the background from the image and get the foreground objects for study. Removing context fixed objects could be used in various

[*] Corresponding Author. Email: karasfi@qiau.ac.ir

algorithms, including video monitoring, optical motion estimation of multimedia applications, conference calls, and computer interfaces.

The purpose of this paper is to reconstruct the images by removing their noises. After reconstruction and using the Mixture of Gaussians (MOG) method, the background removal operation and foreground area separation from the background area perform high. The reason for use MOG is that this method yields better results than similar methods based on previous studies. It is noteworthy that the MOG method will use only to remove fixed elements of the image. If the image has less noise, it can competently remove fixed elements. Directshow framework based on pipe-and-filter architecture uses to remove noises. The proposed approach will test with the median filter and Gaussian filter methods in different scenarios to evaluate efficiency.

In chapter 2, we describe the methods used to remove the background environment. In chapter 3, we provide a formal description of the DirectShow architecture. Chapter 4 introduces MOG. In Chapter 5, we provide details of the proposed method. In Chapter 6, we discuss the datasets and the results. Finally, we present our conclusions in Chapter 7.

2. Literature Review

Background subtraction approaches have been dividing into recursive and non-recursive[3]. Recursive techniques update the background model as new observations arrive, therefore consuming low resources in computational and memory requirements. Examples of this kind of system include the approximated filter method and MOG. On the other hand, non-recursive approaches keep a buffer of the last incoming video frames to estimate the background. Therefore, non-recursive systems have higher memory requirements. Nevertheless, since they have a copy of the most recent video frames, they can cope with some challenges as outlier rejection and fast convergence, which recursive techniques cannot easily handle. Examples of this kind of approach are frame differencing, median filtering, and linear predictive filter.

There are various ways to remove background to identify fixed objects in the scene, each of which confronts challenges such as background storm conditions, occlusion of items, and lighting changes throughout the day. MOG is an effective method for detecting moving objects and is used in complex environments effectively [4, 5].

Shah et al.[5] adopt the MOG as the fundamental framework for their complete system. A self-adaptive method permits an automatic selection of the parameters for the MOG. After this step, they introduce several solutions to address challenges such as ghosts and sudden illumination changes in the environment. They used a voting-based scheme to extract spatial and temporal information to refine the foreground mask. Then, they used the temporal and spatial history of foreground blobs to detect and handle paused objects. Their model shows significant robustness in the presence of ghosts and illumination changes.

Shimada et al.[6] proposed a new framework for the Gaussian mixture models (GMM) to reduce the memory requirement without loss of accuracy. This framework is case-based; this means that the framework removes a background model only when necessary. Furthermore, a case-by-case model share by some of the pixels. Finally, pixel features divide into two groups: the first group is for model selection, and the second group is for modeling. This approach discovers a low-cost and high accurate background model. The memory usage and the computational cost could reduce by half of the traditional GMM with better accuracy.

Alvar et al. [1] presented an algorithm called Mixture of Merged Gaussian Algorithm (MMGA) to reduce the execution time to achieve real-time performance without loss of reliability and accuracy. The algorithm divides into two parts: the probabilistic model of the MOG and the RTDENN model[7]. Results show that the MMGA achieves a significant reduction of execution time compared to the MOG with a higher degree of robustness against noise and illumination changes.

In a MOG, using the threshold value updated in each frame, you could remove the background from the image. In recent years, the SURF feature extraction algorithm has improved the GMM function, which uses this algorithm to remove the context with high precision. The background removal always involves problems such as the complexity of the environment[8]. To solve these problems, we should correctly adjust the values of weight, mean, and variance[9]. Eigen and KDE are suited to remove complex background[10], while these methods are not suitable for real-time applications based on their memory needs.

The researchers concluded that the techniques used at the gray levels have low accuracy than color images, and these techniques in high-noise video scenes will not bring enough precision. Another background removal model, such as the support vector model, is suitable for dynamic background[11]. Other models also include a neural network that provides the right balance between cost and performance.

The other obstacles in intelligent video surveillance systems are limiting the time and checking places where the object is not. With the help of mathematical and computational techniques[12], it is possible to improve the accuracy of video surveillance systems, which have received enough attention in recent years. Optical changes dynamic such as feature selection and hierarchical modeling use to counteract dynamic backgrounds. These models, take from fixed cameras and mobiles, could be helpful strategies [13]. In[14], researchers resolved existing problems in background removal by integrating sequential frames and occlusion management. To correctly classify objects and complex interactions, it is necessary to implement data-matching methods and object tracking in test environments[15]. It must process high-speed video with accurate and quick cameras to identify targets in video monitoring systems. Therefore, a plan should design to increase the processing speed of the video and maintain the accuracy of the target identification.

Data may be streamed and processed in various topologies, including hierarchical, parallel, and tree topology[16]. Deposition of each part of the video processing into a component is one of the advantages of pipe-and-filter architecture. It could create synchronization between different parts of the video processing.

3. Directshow Framework

DirectShow is a multimedia framework provided by Microsoft to perform various operations with media files or data streams. DirectShow divides complex media operations like video playback into sequences of processing steps called filters. Each filter takes on a single stage of data processing and has an input and output that could communicate with another filter.

The communication mechanism is such that filters could communicate in various ways based on a variety of capabilities. To execute a complex task, the developer should first create a graph of filters and then implement the association of the filters together[17]. Available filters include 'source' filter, 'transform' filter, and 'rendering' filters. The 'source' filter, 'transform' filter, and 'render' filter use to read the MP3 file, translate and decode the sound, and run the audio file. Each filter contains pins that could communicate with other filters. Each pin could connect with another plug, and for this, both pins should agree on the transfer data. Most filters are implemented based on C ++ classes.

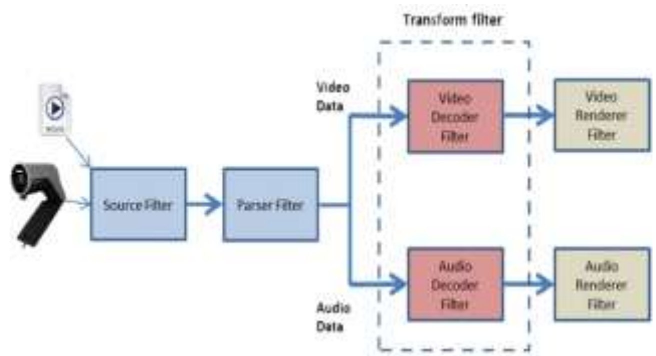

Fig.1. Directshow framework[17].

4. Mixture of Gaussian

First, we apply the background subtraction method to separate moving objects from videos using improved adaptive GMM. Such a method is robust against specific challenges like illumination variance over the day, shadows, shaking tree branches, and other sudden changes. We use a variable number of Gaussian models for each pixel because a single Gaussian is not sufficient to completely model these variations in complex and varying situations. Here we provide a brief overview of the improved adaptive Gaussian mixture model.

Suppose that I^1, I^2, …, I^t is the intensity of a pixel for past t consecutive frames. Then at the time, the probability of observing the current pixel value is:

$$P(I^t) = \sum_{i=1}^{k} w_i^t \ * \eta(I^t . \mu_i^t . \Sigma_i^t) \tag{1}$$

Where k is the number of distributions, w_i^t is weight, and $\eta(I^t, \mu_i^t, \Sigma_i^t)$ is i^{th} Gaussian probability density function with mean μ_i^t and Σ_i^t as variance at time t. the available memory and computational power determine k.

For each pixel, the Gaussian components with low friction and high weight correspond to the background class, and others with high variance correspond to the foreground class[18]. At time t, the pixel intensity I^t checks against all Gaussian components. If i^{th} component satisfies the condition:

$$|\mu_i^t - I^t| < \beta_i \Sigma_i \tag{2}$$

Then i^{th} element is considered to be a match. Also, the current pixel is classified as background or foreground according to the class of i^{th} Gaussian model. The prior weights of the k distributions at time t, w_i^t, are adjusted as follows:

$$w_i^t = (1-\alpha) w_i^{t-1} + \alpha(\mathrm{M}_i^t) \tag{3}$$

Where α is the learning rate determines how frequently parameters are adjusted, and M_i^t is 1 for the model that matched and 0 for the remaining models. After this approximation, the weights renormalize. Here, β_i is a threshold that has a significant impact when different regions have different Lightning. Generally, the value of β_i is kept around 3, as $\mu^t \pm 3\Sigma_i^t$ accounts for approximately of data. The parameters of the distribution which matches the new observation are updated as follows:

$$\mu_i^t = (1-\rho)\mu_i^{t-1} + \rho I^t \tag{4}$$

$$(\Sigma_i^t)^2 = (1-\rho)(\Sigma_i^{t-1})^2 + \rho(I^t - \mu_i^t)^2 \tag{5}$$

Here, $\rho = \eta\ (I^t \mid \mu_i, \Sigma_i)$. A new Gaussian model is created with the current pixel value as mean, low prior weight, and high variance when there is no matched component. This newly created model replaces the least probable component or adds as a new component if the maximum number of components reaches or not, respectively.

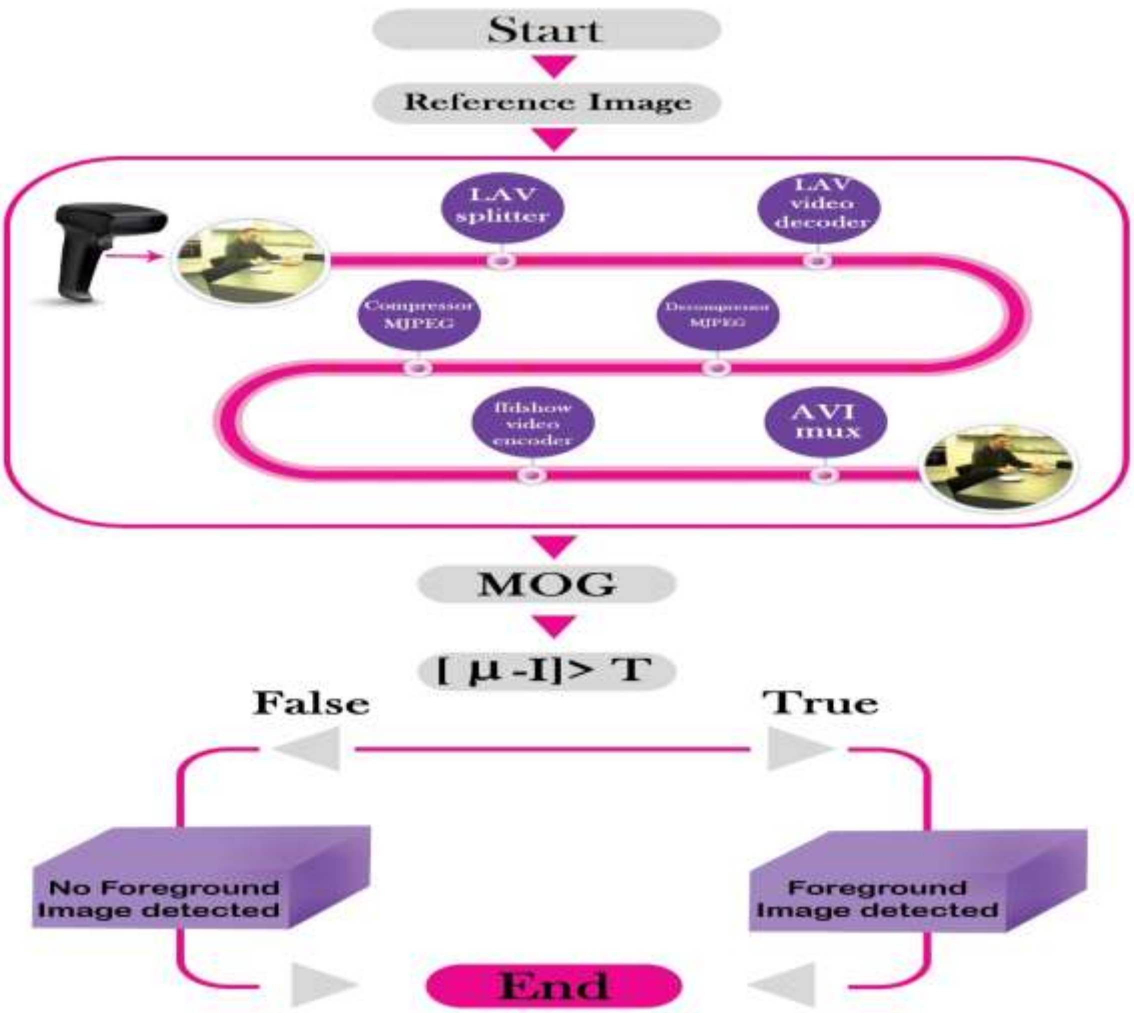

Fig.2. Diagram of the proposed method.

5. Implementation Directshow Filter for Existing Noise Reduction In Images

As mentioned, the most critical challenge in removing background objects is noise in the images. Therefore, an approach should be adopted that separates background objects from the foreground by removing noise in the photos. For this purpose, the Directshow framework based on pipe-and-filter architecture uses to removes existing noise in photos with operations such as compression, encryption, decryption, etc.

The reason for use pipe-and-filter architecture is that each component plays a part in processing operations, which causes increased processing operations at high speed and precision. In the Directshow framework, each piece could not communicate with another piece, and only the components of the same type[19] could communicate with each other. In this section, using filtering in Directshow, first, one should perform compression and encoding operations on the video to reduce noise and then implement the background removal using the MOG. So, first, the components which are used in the Directshow filter to minimize the error are checked. According to Fig.2, the parts used are:

1) LAV splitter: This component divides audio and video streams into AVI file playback.
2) LAV Video decoder: This component performs decoding in LAV format.

3) Compressor MJPEG: Compresses the component of the LAV format into JPEG format.
4) Decompressor MJPEG: After compression, this component is ready to perform processing in the 'ffdshow' part.
5) The 'ffdshow' Video encoder: This component supports various formats, including Xvid H.264 and DivX. This component improves video quality by using resizing and sound quality using re-sampling.
6) AVI mux: This component integrates multiple data streams into AVI format. This component considers one input for each input stream. It will show in later sections that the Directshow filter will change the error criteria, including SNR, PSNR, MSE, and processing time.

In Fig. 2, after applying the Directshow filter on the image, as mentioned in the introduction, the MOG method will use to remove fixed elements of the picture. If the image has high noise, it can't operate removing fixed elements and then can't detect the moving objects of the image with high accuracy. So here we can understand the critical role of the Directshow filter.

6. Results and Discussions

In this section, the results of the experiments compare with other methods. Matlab is used to simulate and estimate the error criteria.

6.1. Dataset

Three sets of data use for testing. In the first experiment, the testing video is from an office environment where employees are walking. In the second experiment, the person is cleaning the table. Finally, in the third experiment, the person is peeling off the vegetables. In table 1, you could see info about datasets.

Table 1
Details of datasets.

Dataset	Frame count	Time	Used frames
Office Environment	330	12s	5, 10, 45, 80, 100, 150, 200, 250 and 300
KIT Robo Kitchen(the person is cleaning the table)[20]	1111	74s	5, 10, 45, 80, 100, 150, 200, 250 and 300
KIT Robo Kitchen(the person is peeling off the vegetables)[20]	1752	116s	5, 10, 45, 80, 100, 150, 200, 250 and 300

6.2. Experimental Result

The results show that the method used is far superior to existing methods, and the calculated values for the MSE error criteria and the processing time have significantly reduced. There are high values for the SNR and PSNR criteria using the Directshow filter, indicating high-quality image restoration. The experiments implement in three steps.

6.2.1. Experiment1

In this step, we add the salt and pepper noise and the Gaussian noise to the image and then applying a median filter, Gaussian filter, and Directshow filter to the picture. The comprehensiveness is the reason for using these noises. It is noteworthy that the mean and variance values for Gaussian noise are 0.2 and 0.01, respectively. Also, the noise density value for salt and pepper noise set 0.2.

6.2.1.1. Discussion

According to the data in table 2 and table 3, one can conclude that if the video quality is moderate in eliminating salt and pepper noise, the effects of the proposed method on the SNR, PSNR, and MSE values will be upper.In frame5, the MSE amount in a low-quality video is reduced by 16 units while decreasing by an average of 76 ones in an average quality video. Also, the SNR amount and the PSNR value amount will increase by six units and three units in a low-quality video, respectively. In contrast, in a mediate quality video, the SNR and PSNR values are increased by 30 unities and 16 unities.The lower and higher quality of the video, the difference in PSNR values and SNR values will be less than other methods. In frame 45 of the first and third videos, the amount of MSE will significantly reduce, and in the first and third videos, the PSNR and SNR values will dramatically increase. It is noteworthy that video 3 has more quality than video 2, and video 2 has more than video 1.

Table 2
Comparison of MSE, PSNR, and SNR values in the first, second, and third video in the presence of salt and pepper noise and applying median and Directshow filter (M.F.: Median filter, PM: DF: Directshow filter).

Frame	First video						Second video						Third video					
	MSE		SNR		PSNR		MSE		SNR		PSNR		MSE		SNR		PSNR	
	M.F	D.F	M.F	D.F	M.F	D.F	M.F	D.F	M.F	D.F	M.F	DF	M.F	DF	M.F	DF	M.F	DF
5	30.01	**13.58**	18.58	**25.29**	33.39	**36.75**	78.34	**1.52**	10.25	**40.55**	29.23	**46.34**	99.20	**62.95**	8.20	**12.15**	28.20	**30.18**
10	27.86	**13.21**	19.23	**25.71**	33.72	**36.96**	77.82	**1.50**	10.30	**44.58**	29.25	**46.39**	99.61	**63.18**	8.16	**12.11**	28.18	**30.16**
45	29.37	**14.98**	18.77	**24.61**	33.49	**36.41**	53.43	**0.94**	13.57	**48.69**	30.89	**48.44**	100.0	**63.76**	8.13	**12.03**	28.16	**30.12**
80	31.03	**15.24**	18.29	**24.46**	33.25	**36.33**	56.77	**1.07**	13.04	**47.55**	30.62	**47.87**	100.50	**63.74**	8.08	**12.04**	28.14	**30.12**
100	29.13	**13.68**	18.84	**25.41**	33.52	**36.80**	74.47	**1.46**	10.69	**44.81**	29.45	**46.51**	99.62	**64.25**	8.16	**11.97**	28.18	**30.09**
150	29.14	**14.65**	18.84	**24.81**	33.52	**36.51**	64.24	**1.50**	11.97	**44.61**	30.09	**46.41**	100.83	**63.25**	8.05	**12.10**	28.13	**30.15**
200	30.29	**13.80**	18.50	**25.33**	33.35	**36.77**	74.54	**1.63**	10.68	**43.91**	29.44	**26.05**	100.67	**63.50**	8.07	**12.07**	28.14	**30.14**
250	29.60	**12.90**	18.70	**25.92**	33.45	**37.06**	71.99	**1.52**	10.98	**44.49**	29.59	**46.35**	100.17	**64.35**	8.11	**11.96**	28.16	**30.08**
300	28.61	**13.69**	18.99	**25.40**	33.60	**36.80**	72.76	**1.50**	10.89	**44.60**	29.55	**46.40**	100.65	**62.95**	8.07	**12.15**	28.14	**30.17**

Table 3
Comparison of MSE, PSNR, and SNR values in the first, second, and third video in the presence of Gaussian noise and applying Gaussian filter and Directshow (G.F.: Gaussian filter, D.F: Directshow filter).

Frame	First video						Second video						Third video					
	MSE		SNR		PSNR		MSE		SNR		PSNR		MSE		SNR		PSNR	
	G.F	D.F	G.F	D.F	G.F	D.F	G.F	D.F	G.F	D.F	G.F	D.F	G.F	D.F	G.F	D.F	G.F	D.F
5	83.03	**25.95**	9.74	**19.84**	28.97	**34.02**	98.82	**81.30**	8.23	**9.92**	28.22	**29.06**	112.51	**65.41**	10.21	**11.81**	27.65	**30.01**
10	84.05	**26.28**	9.64	**19.73**	28.92	**33.97**	98.91	**80.46**	8.22	**10.01**	28.21	**29.11**	113.09	**65.33**	7.06	**11.82**	27.63	**30.01**
45	83.53	**25.44**	9.69	**20.02**	28.95	**34.11**	86.72	**26.83**	9.36	**19.55**	28.78	**33.88**	113.18	**66.14**	7.05	**11.72**	27.63	**29.96**
80	84.11	**26.23**	9.63	**19.75**	28.92	**33.98**	88.44	**39.87**	9.19	**16.11**	28.70	**32.16**	113.45	**66.64**	7.03	**11.65**	27.62	**29.93**
100	84.13	**26.99**	9.63	**19.50**	28.92	**33.85**	96.71	**74.02**	8.42	**10.74**	28.31	**29.47**	113.31	**66.40**	7.04	**11.68**	27.62	**29.94**
150	83.60	**26.68**	9.68	**19.60**	28.94	**33.90**	91.49	**66.34**	8.90	**11.69**	28.55	**29.95**	113.49	**65.88**	7.09	**11.75**	27.62	**29.98**
200	84.64	**26.73**	9.57	**19.59**	28.89	**33.89**	96.59	**78.09**	8.43	**10.27**	28.32	**29.24**	113.52	**66.41**	7.02	**11.68**	27.61	**29.94**
250	83.83	**25.43**	9.66	**20.02**	28.93	**34.11**	97.56	**77.85**	8.34	**10.30**	28.27	**29.25**	113.37	**67.13**	7.04	**11.59**	27.62	**29.90**
300	83.24	**26.25**	9.72	**19.74**	28.96	**33.97**	98.09	**78.15**	8.29	**10.27**	28.25	**29.24**	113.27	**65.79**	7.04	**11.76**	27.62	**29.98**

Allows the conclusion that if the video quality is higher and lower in eliminating Gaussian noise, the effects of the proposed method on the SNR, PSNR, and MSE values will be upper. In frame five, the MSE value in a moderate quality video is reduced by 18 units while decreasing by an average of 47 ones in a higher quality video. Also, the SNR of 3 ones and the PSNR of 1 in a moderate quality video will increase. In contrast, in a higher quality video, the SNR and PSNR values will increase four and three ones, respectively. With the moderate quality of the video, the difference in PSNR and SNR values presented in the method will be less than other methods. For example, in frame 45 of the second video, the amount of MSE will significantly reduce, and in the second video, the PSNR and SNR values will not increase substantially. In Fig. 3, You could see the comparison of the average MSE, SNR, and PSNR. In Fig. 4, you could see the results of applying the Directshow filter in Gaussian noise and salt and pepper noise.

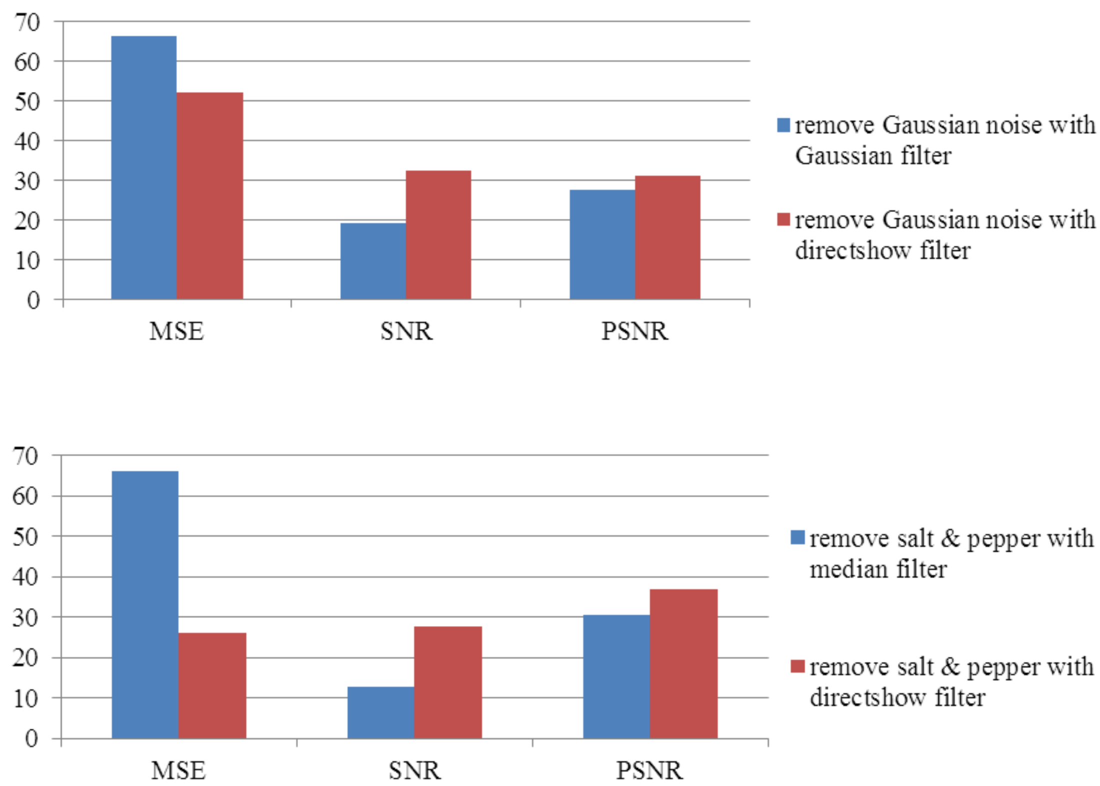

Fig.3. Comparison of the average MSE, SNR, and PSNR: in the frames of first, second, and third video. Top) Comparison of Gaussian noise removal in Gaussian and Directshow filters. Bottom) Comparison of salt and pepper noise removal in median and Directshow filters.

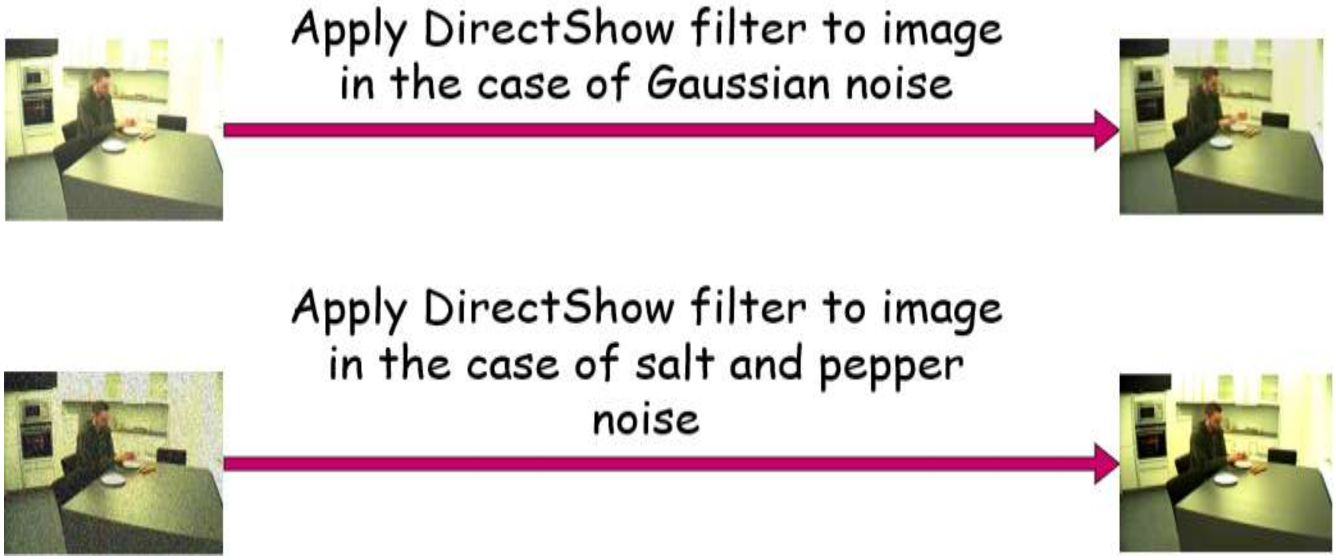

Fig.4. Results of applying the Directshow filter in case of Gaussian noise and salt and pepper noise

6.2.2. Experiment2

We calculate processing time in applying and not applying the Directshow filter to the image in this step. According to Fig. 5, the Gaussian and salt and pepper noises first add to the video. Then the processing time for noise removal is calculated by Directshow filter, Gaussian filter, and median filter.

6.2.2.1. Discussion

In this step, as you could see in Fig.4, the processing time is reduced because of image restoration in using the Directshow filter. The frame rate for the first, second, and third videos is 17.25, 8.69, and 8.70, respectively.

The processing time for 30 frames and 60 frames is 33 ms and 60 ms, respectively. In this study, the processing time is lower (17.26, 8.69, 8.70) for the desired frames that indicate faster processing using the Directshow filter.

Given that computations implement in Matlab, the frame rate could increase by two to three times by increasing the processing in other environments. Finally, as a result, Directshow filters could be used in real-time environments. The reason for improving the system's real-time performance in using Directshow filters is that it could process frames at the right time and with high quality. You could see the process of noise removal with the median filter, Gaussian filter, and Directshow filter in Fig.5 and the results of the processing time in table 4.

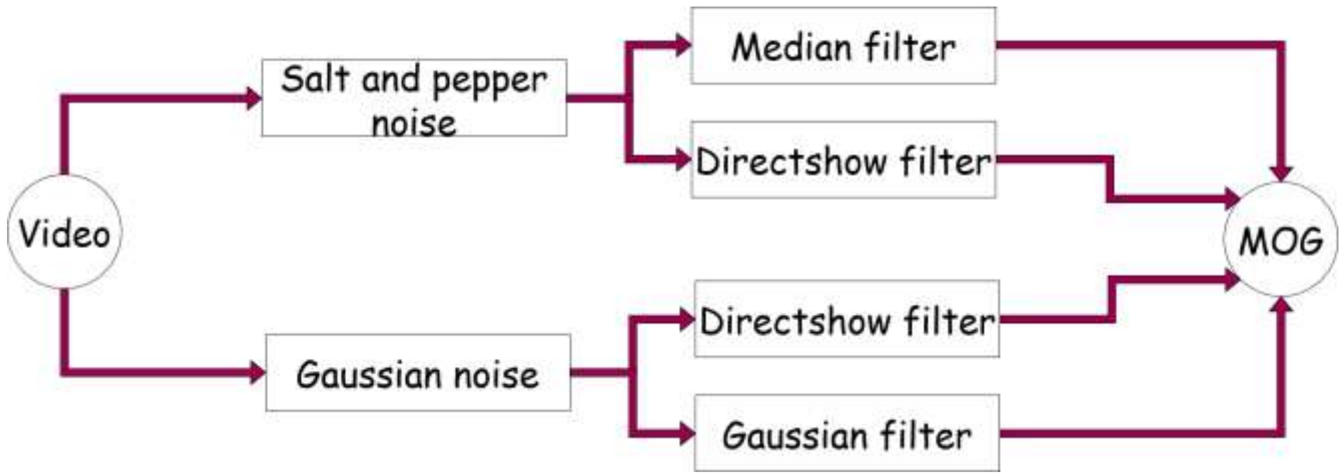

Fig.5. checking processing time using median filter, Gaussian filter, and Directshow filter in case of existing Gaussian noise and salt and pepper noise.

Table 4
Comparison of processing time (per second) using median filter, Gaussian filter, and Directshow filter.

Video	Median filter	Gaussian filter	Directshow filter
First	24.45	23.68	**19.13**
Second	247.21	134.90	**127.78**
Third	374.50	212.83	**201.29**

6.2.3. Experiment3

In this step, we show the effect of applying the Directshow filter to video to extract video foreground. In other words, the background subtraction operation must take place by using Gaussian, median, and Directshow filters without noise images, and finally, the error criteria compare.

6.2.3.1. Discussion

In this step, we will calculate the error criteria for the state that we want to get the foreground image. In the first scenario, we obtain the foreground image when a Gaussian filter applies to the video and then calculate the MSE, SNR, and PSNR values. In the second scenario, we obtain the foreground image when the median filter applies to the video and then calculate the MSE, SNR, and PSNR values. In the third scenario, we obtain the foreground image when a Directshow filter applies to the video and then calculate the MSE, SNR, and PSNR values. You could see the comparison for the first, second, and third videos in Fig.6, 7, and 8.

As shown in these figures, the SNR and PSNR values for the Directshow filter are higher than the Gaussian filter, and the SNR and PSNR values for the Gaussian filter are higher than the median filter. Also, the MSE value of the Directshow filter is lower than the Gaussian filter, and the MSE value of the Gaussian filter is lower than the median filter.

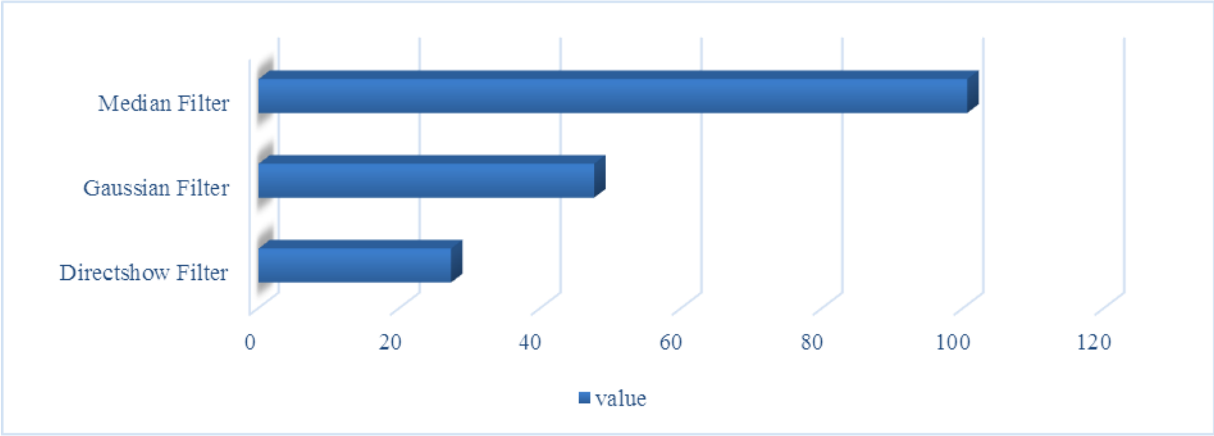

Fig.6. Calculate the MSE value in the foreground image for the first video.

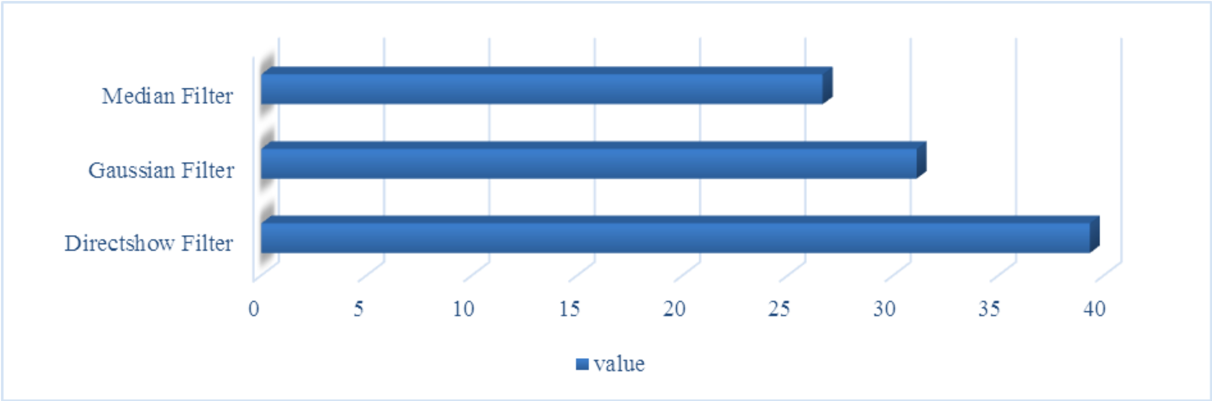

Fig.7. Calculate the SNR value in the foreground image for the second video.

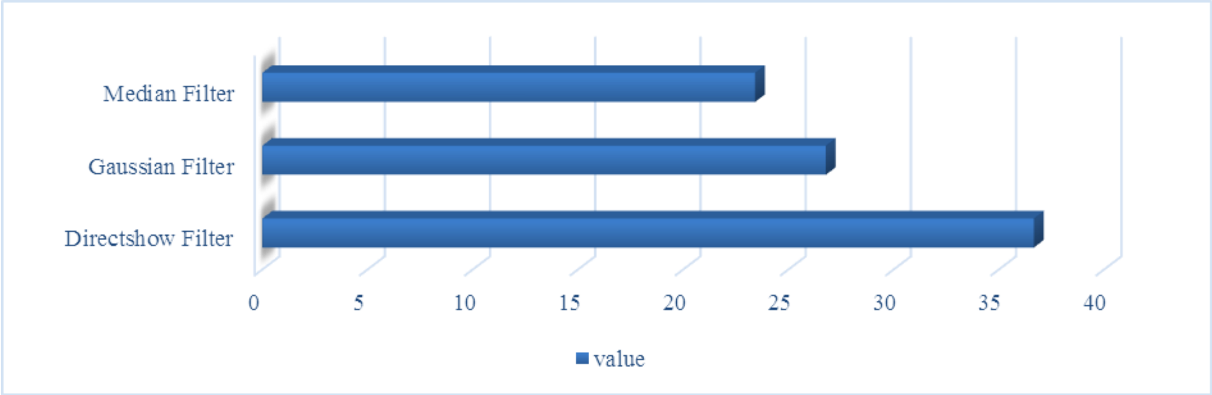

Fig.8. Calculate the PSNR value in the foreground image for the third video.

7. Conclusions

In this paper, regarding the obtained values, one can conclude that the MSE, SNR, and PSNR error criteria for the conditions that the MOG used with the Directshow filter for background removal are much better than the state only MOG used. It means that by applying the Directshow filter to the video, the observed noise reduction, and with very high performance, it is possible to improve the MOG for background removal and detect and track the target in the video. The Directshow filter uses a pipe-and-filter architecture to process operations at high speed and precision.

References

[1] H. Ji and et al., Robust video denoising using low rank matrix completion. in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1791-1798, 2010.

[2] H.R.Schmidtke, A survey on verification strategies for intelligent transportation systems. Journal of Reliable Intelligent Environments, vol. 4, no. 4, pp. 211–224, 2018.

[3] K.Vignesh, G. Yadav and A.Sethi, Abnormal Event Detection on BMTT-PETS 2017 Surveillance Challenge. Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2161-2168, 2017.

[4] A. Jalalian, B. Karasfi, K. Samsudin, M. I. Saripan and S. Mashohor, Fast Cellular Automata Implementation on Graphic Processor Unit (GPU) for Salt and Pepper Noise Removal. Journal of Computer & Robotics, vol. 7, no. 2, pp. 21–28, 2014.

[5] M. Shah, J. D. Deng and B. J. Woodford, Video background modeling: recent approaches, issues and our proposed techniques. Machine Vision and Applications, vol. 25, no. 5, pp. 1105-1119, 2014.

[6] H. P. Donovan and S. F. Sidney, Evaluation of Background Subtraction Algorithms with Post-Processing. 2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance, pp. 192-199, 2008.

[7] A. Shimada, Y. Nonaka, H. Nagahara and R. Taniguchi, Case-based background modeling: associative background database towards low-cost and high-performance change detection. Machine Vision and Applications, vol. 25, no. 5, pp. 1121–1131, 2014.

[8] M. Alvar, A. Rodriguez-Calvo, A. Sanchez-Miralles and A. Arranz, Mixture of Merged Gaussian Algorithm using RTDENN. Machine Vision and Applications, vol. 25, no. 5, pp. 1133-1144, 2013.

[9] Y. Xu, J. Donga, B. Zhang and D. Xu, Background modeling methods in video analysis: A review and comparative evaluation. CAAI Transactions on Intelligence Technology, vol. 1, no. 1, pp. 43-60, 2016.

[10] K. Goyal and J. Singhai, Review of background subtraction methods using Gaussian mixture model for video surveillance systems. Artificial Intelligence Review, vol. 50, no. 2, pp. 1-19, 2018.

[11] S. Joudaki, M. S. Bin Sunar and H. Kolivand, Background Subtraction Methods in Video Streams: A Review. 4th International Conference on Interactive Digital Media (ICIDM), pp. 1-6, 2015.

[12] T. ouwmans, Traditional and recent approaches in background modeling for foreground detection: An overview. Computer Science Review, vol. 11, pp. 31-66, 2015.

[13] S. Gruenwedel, N. I. Petrovic, L. Jovanov, J. O. Nino-Casta-neda, A. Pizurica and W. Philips, Efficient foreground detection for real-time surveillance applications. Electronics Letters, vol. 49, no. 18, pp. 1143-1145, 2013.

[14] T. Bouwmans, J. Gonzalez, C. Shan, M. Piccardi and L. Davis, Special issue on background modeling for foreground detection in real-world dynamic scenes. Machine Vision and Applications, vol. 25, no. 5, pp. 1101–1103, 2014.

[15] A. Bayona, J. C. SanMiguel and J. M. Martínez, Stationary foreground detection using background subtraction and temporal difference in video surveillance. Image Processing (ICIP), pp. 4657-4660, 2010.

[16] M. Valera and S. A. Velastin, Intelligent distributed surveillance systems: a review. IEE Proceedings - Vision, Image and Signal Processing, vol. 152, no. 2, pp. 192-204, May 2005.

[17] B. Babcock, S. Babu, M. Datar, R. Motwani and J. Widom, Models and issues in data stream systems. Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 1-16, June 2002.

[18] M. Linetsky, Programming Microsoft Directshow. 2001.

[19] Z. Zivkovic, Improved adaptive Gaussian mixture model for background subtraction. Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 2, pp. 28-31, 2004.

[20] M. Wojnarski, Debellor: A Data Mining Platform with Stream Architecture. Springer Berlin Heidelberg, vol. 9, pp. 405-427, 2008.

[21] L. Rybok, S. Friedberger, U. D. Hanebeck and R. Stiefelhagen, The KIT Robo-kitchen data set for the evaluation of view-based activity recognition systems. 2011 11th IEEE-RAS International Conference on Humanoid Robots, pp. 128-133, 2011.