Moving Human Target Detection and Tracking in Video Frames

Manikandaprabu NALLASIVAM[1]*, Vijayachitra SENNIAPPAN[2]

[1] Nandha Engineering College, Erode, 638052, Tamil Nadu, India
manikandaprabube@gmail.com (**Corresponding author*)

[2] Kongu Engineering College, Erode, 638052, Tamil Nadu, India
dr.svijayachitra@gmail.com

Abstract: The conventional method for moving human target detection and tracking has come across a major setback due to various hindering factors such as environmental lighting conditions, temperature, etc. Similarly, it has been noticed that the manual selection of moving human targets in a video sequence does not provide convincing results either. In this paper, a new method for moving human target detection and tracking is proposed. It involves two stages. The first stage consists in the detection of moving human targets and the second one in target tracking based on the Continuously Adaptive Mean-Shift (CAMShift) algorithm. In the first stage, in order to select the moving target, the background subtraction method and frame subtraction method are combined. The Region Of Interest (ROI), which is usually the moving target is identified. In the second stage, target tracking is performed by choosing a centroid pixel point over the ROI, which is then used by the CAMShift algorithm. The proposed method has shown outperforming results for various performance parameters such as precision, accuracy, recall, and the F1-score under three different lighting conditions. The results obtained also show a reduction in time complexity in comparison with the state-of-the-art algorithms.

Keywords: Background subtraction, Frame subtraction, CAMShift algorithm, Target detection, Target tracking.

1. Introduction

Surveillance video systems are being increasingly used for everyday security. Surveillance cameras are available with various resolutions. The frame rate of each camera differs based on the resolution. Usually, the frame rate for a camera of average quality (1280 x 720) is around 30 fps. In an automated video surveillance system, moving target detection and tracking of dynamic circumstances remains a challenging task. For the detection of a moving target, a fixed background model is necessary, from which the foreground moving target can be extracted.

On the other hand, the foreground moving target may be extracted based on various features such as color, shape, edges, texture, etc. Trainable classifiers play a major part in automated systems and have proved to work efficiently in target detection with recorded video databases (Rahmaniar, Wang & Chen, 2019). The major drawback of the trainable classifiers is that the computational time for processing each frame exceeds the average frame rate of a camera (Guerrero-Ibáñez, Zeadally & Contreras-Castillo, 2018). Similarly, the application of a deep learning algorithm for this type of problem has become too expensive due to the need for a dedicated Graphical Processing Unit (GPU), and the database framing time and training time are also high when compared with the proposed algorithm (Algabri & Choi, 2020; Dou, Qin, & Tu, 2019).

These drawbacks are avoided by designing a simple, efficient algorithm for live monitoring in the automated surveillance system.

This paper is organized as follows. Section 2 presents the related works. Section 3 sets forth, the target detection process using background subtraction and frame subtraction methods. Section 4 describes the target tracking process using the CAMShift algorithm. Section 5 includes the experimental analysis and the related results. Finally, the conclusion is presented in Section 6.

2. Related work

Akli et al. (2021) proposed an active contour-based moving object detection and Kalman filter-based tracking with camera motion compensation. The active contour algorithm detects the exact boundary of the moving object and the detected object was tracked by Kalman filter. The implementation of Smooth Variable Structure Filter (SVSF) improved the robustness in motion estimation. Video stabilization achieved by using a homography matrix, reduces the unwanted camera vibrations, shakes, and motion blur. Similarly, another approach based on the Kalman filter and fusion of multi-resolution features was proposed by (Zhou & Zhang, 2019), it provides a solution for overcoming the shortcomings in the Siamese network. The Kalman filter was used

https://doi.org/10.24846/v30i1y202111

for tracking the complex scenes and provides stability in tracking the fast-moving objects. The online learning process of the system along with the fusion of multi-resolution features provides a multiple response score map and supports the system in tracking a variety of targets.

(Ray & Chakraborty, 2017a; Ray & Chakraborty, 2019) proposed an object detection and tracking method for dynamic backgrounds in video frames. For estimation and compensation of pseudo-motion in the background, phase correlation of consecutive frames based on Fourier Shift Theorem was used. The moving object was detected by using background subtraction method, by computing the difference between background model and current frame. The detected object was refined by morphological filters and tracked by the Kalman filter (Cheok, Omar, & Jaward, 2019). Similarly, another approach based on background subtraction algorithm and deep neural network for improving human detection rate in surveillance videos was proposed by Kim et al. (2018). This framework was designed to detect and recognize the moving targets in outdoor Closed Circuit Tele-Vision (CCTV) surveillance videos. A two-stage process was involved, initially performing region of interest (ROI) detection by using background subtraction algorithm in the first stage, followed by Convolutional Neural Network (CNN) classification for recognizing the predefined classes in the second one. Despite achieving an accuracy of 85%, the computational time for processing each frame was increased, which is a major drawback for the trainable classifiers. (Ray & Chakraborty, 2017b) proposed an algorithm for moving camera-based object detection and tracking. In this method, both the background and foreground objects are changing in every frame. The moving objects are identified by generating spatio-temporal blobs in every frame using a 3D Gabor filter. The minimum spanning tree was employed in order to merge the individual blobs to locate and track the moving target by the values of certain features such as height, width, histogram bin value, etc. These extracted feature values are used for generating the trajectories of the moving object in the video frames. The Kalman filter was used to handle the problem of occlusion in video frames (Kim et al., 2018).

(Cao et al., 2017) proposed a Motion–Adaptive Particle Filter (MAPF) for complex dynamics to overcome the abrupt motions and affine transformation problems in object tracking. A Sub-Particle Drift (SPD) approach was used to predict the acceleration and velocity of the target. The migrating target's velocity and propagation distance were calculated with motion estimation and SPD. The result of this approach seems to be same as that of the CAMShift algorithm, where the object moves with high random velocities and acceleration (Chen, Wang & Xia, 2019).

(Ren et al., 2016) proposed a Region Proposal Network (RPN) based on Region-Based Convolutional Neural Networks (R-CNN) which was computationally expensive. The Region Proposal Network is a fully convolutional network that simultaneously predicts the object on each position on the frame. It was fully trained to produce high-quality region proposals which are further used by Faster R-CNN for object detection. Further, the Faster R-CNN and RPN are merged to form a single network by sharing the convolutional features. In this system, the maximum frame rate of 5fps was achieved with a GPU.

Redmon et al. (2016) introduced YOLO9000, a state-of-the-art system, that is a real-time object detection system that can detect over 9000 object categories. The YOLO (You Only Look Once) system was further improved by deploying multi-scale training for the reduction of false detection rate. Most object detection algorithms are designed with a limited number of object categories. This algorithm allows the user to train and detect more object categories quickly and accurately. The requirement to use the most powerful GPU for simulation was the major problem in the real-time implementation of this algorithm in surveillance systems (Redmon & Farhadi, 2017; Redmon et al., 2016).

(Zhao et al., 2016) addressed some issues related to visual tracking, such as motion blur, overlapping, pose changes, similar color distribution in the background, and illumination changes. The background and foreground separation was the initial task in moving object detection (Chrysos et al., 2018; Thabet et al., 2020). The structural local sparse representation methodology was used for locating the moving target in the background region. To face local optimization problems, a weighted search-based methodology was used for refining the located human target in complex background scenarios (Sultana et al., 2020; NaNa et al, 2020). For tracking the identified target, CAMShift searching was used and it results in robustness towards occlusions, appearance changes, and complex backgrounds (Ranjan et al., 2017; Bay et al., 2006).

3. Algorithms for Target Detection in Video Frames

The images captured using surveillance cameras are color images in nature during day vision and grayscale images in night vision. Processing the different nature of images is important in detecting and tracking the moving target. The captured color images are converted to grayscale images by eliminating hue and saturation while retaining the luminance.

3.1 Target Area Detection Using Background Subtraction Method

Background subtraction mainly consists in generating a stable background model for detecting and tracking the moving target by obtaining the complete foreground features. The current frame image is subtracted from the background model developed by the background subtraction method. The absolute difference is taken between background frame and current frame, by fixing the threshold value (Borji et al., 2019). If the subtracted value of the pixel is lower than the threshold value, then it becomes a part of the background image. If the subtracted value of the pixel is higher than the threshold value, then it belongs to the foreground target area. Hence the updating of background model is an important process (Bouwmans et al., 2018).

3.1.1 Establishment and Updating of Background Model

The statistical analysis of continuous video frames for establishment of background model is expressed in equation (1):

$$Bg_n(x,y) = \frac{1}{N}\sum_{i=1}^{N} img_i(x,y) \quad (1)$$

where background image is denoted by $Bg_n(x,y)$ and the number of frames in a video sequence is denoted by N. The gray value of the pixel of coordinates (x,y) of the first frame is denoted by $img_i(x,y)$. The background image is obtained by changing the values of x and y. The background model can be updated using equation (2).

$$Bg_{n+1}(x,y) = aBg_n(x,y) + (1-a)D_n(x,y) \quad (2)$$

where the rate of updating is denoted by a, and its value varies between 0.5 and 1. The background image value at the current moment is $Bg_n(x,y)$, the gray value of the current image is $D_n(x,y)$ and the updated background value is $Bg_{n+1}(x,y)$.

3.1.2 Foreground Image Extraction

Frames are selected based on the brightness variation between foreground and background images. The absolute difference between the current frame and background image is taken and compared to the threshold value T_n. For getting foreground object the values higher than the threshold T_n are set to 1, while those lower than the threshold T_n are set to 0 as background pixels, where foreground image is denoted by $G_n(x,y)$.

$$G_n(x,y) = \begin{cases} 0 & |D_n(x,y) - B_n(x,y)| < T_n \\ 1 & |D_n(x,y) - B_n(x,y)| \geq T_n \end{cases} \quad (3)$$

The above operation supports extracting the binary values. Moving objects can be extracted very quickly by using background subtraction method (Ahmed et al., 2018).

3.2 Target Detection Using Frame Subtraction Method

In the frame subtraction method, the absolute difference is taken between consecutive odd and even frames, which provides an outline of the foreground target. The absolute difference between consecutive odd and even frames is stated in equation (4).

$$D_k(x,y) = |I_k(x,y) - I_{k-1}(x,y)| \quad (4)$$

where the absolute difference between the gray values of consecutive frames k-1 and k is denoted by $D_k(x,y)$. The gray value of even frame and odd frame is denoted by $I_k(x,y)$ and $I_{k-1}(x,y)$, where the binary values are obtained by binarizing $D_k(x,y)$.

$$G_n(x,y) = \begin{cases} 1 & D_k > T_n \\ 0 & D_k \leq T_n \end{cases} \quad (5)$$

where T_n is the threshold value. The value of a pixel is 1, when there is a change of position between the two frames and the value of the pixel is 0, when there is no change of position, through which the required foreground moving targets are extracted. Hence the background subtraction and frame subtraction algorithm are combined for a better result, whereas the target area is obtained by combining background subtraction and image subtraction method.

In Figure 1, the sequential video frames are considered for the gray variation process and a linear filter such as Gaussian filter is used for the background subtraction algorithm. The absolute difference is taken between background frame and current frame and the output is binarized with the threshold values to obtain the approximate moving target. The appropriate moving target is identified by using mathematical morphological operations. A similar process is carried out in frame subtraction method where the filtering process using the linear filters is not necessary as it is the case with the background subtraction method. The grayscale converted image is directly considered as odd frames and even frames by which the absolute difference is calculated, followed by binarization and mathematical morphological operations. The final images obtained by using background subtraction method and frame subtraction method are combined to detect the exact moving target.

4. CAMShift Algorithm Based Target Tracking in Video Sequences

4.1 Moving Target Tracking

The upgraded version of the mean shift algorithm is used for tracking the probability distribution of varying features. Initially, the matching template is obtained from the probability distribution of color using histogram back projection. The mean shift algorithm is applied to the continuous video frames and the data obtained is adjusted adaptively (Xia et al., 2018). The centroid of the moving target in the current video frame can be computed based on the search window size and location (Naji et al., 2019).

Back projection: The color histogram of the hue component is obtained after uniform quantization from the video frames converted

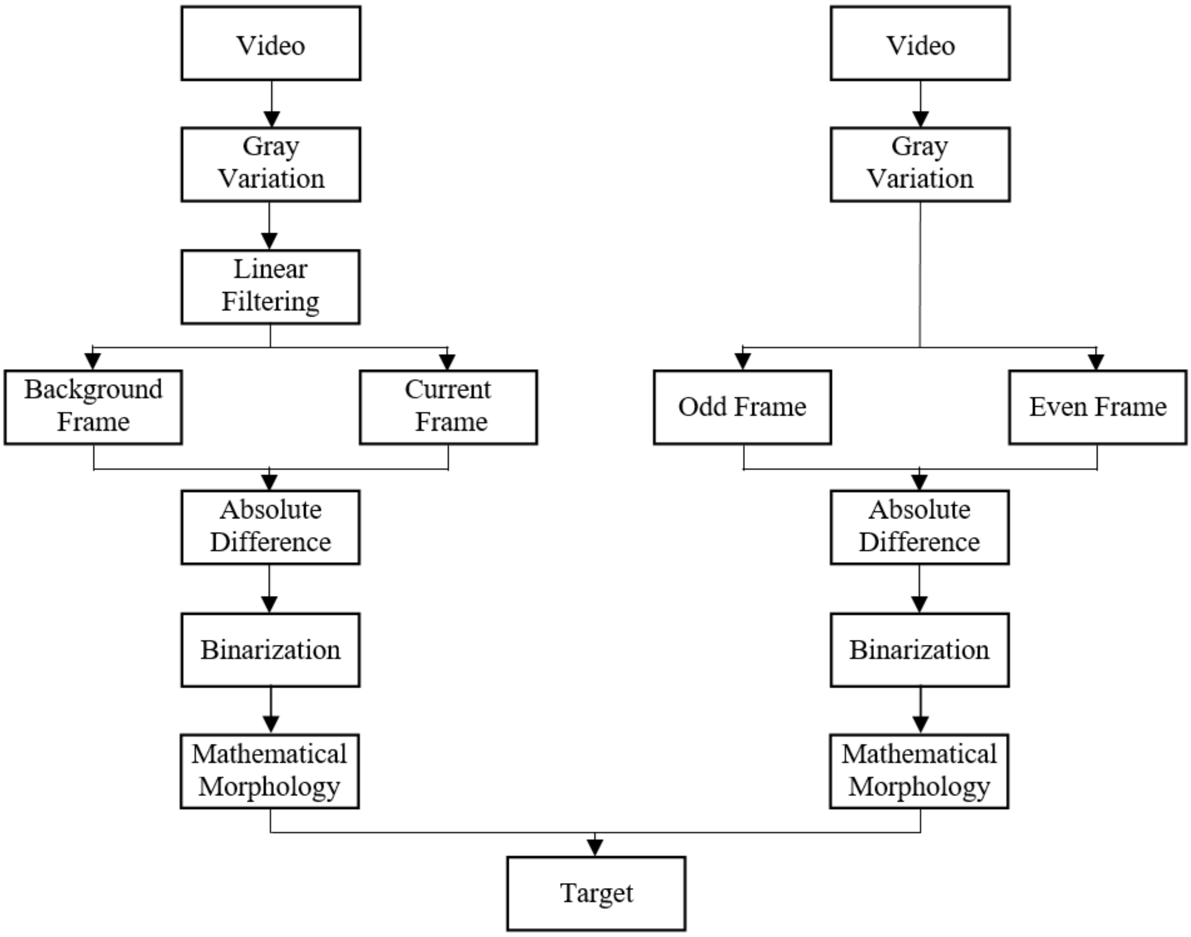

Figure 1. The combined background subtraction method and frame subtraction method

from RGB (Red Green Blue) to HSV (Hue Saturation Value) color space. The histogram is calculated based on the probability of color occurrence from the lookup table by which, the coordinate color values are replaced.

Mean shift iteration: The centroid value of the moving target is compared to the threshold value for determining if the moving distance has exceeded it (Sliti et al., 2018). If the determined target's moving distance exceeds the threshold value, then the center of mass, area, and position of the window is adjusted and the new results are calculated. If the moving distance is lower than the threshold value, then the target is repositioned in the next frame by calculating the convergence condition (Daku et al., 2019; Manikandaprabu & Vijayachitra, 2019).

CAMShift: The upgraded version of the mean shift algorithm for continuous video frames is the CAMShift algorithm. The initial values of the location and size of the current frame are adjusted based on the values obtained by applying the mean shift algorithm on the consecutive frame. By performing this, the location of the moving target is obtained and continuous tracking of the moving target is achieved.

4.2 Combined Frame Subtraction and Background Subtraction Method

The tracking algorithm mainly depends on the gray-level details of the target. When the size and orientation of the target change, it is very difficult for the mean shift algorithm to continuously track the target perfectly. The color feature remains stable and the effect of changes in light and target size is diminished. The RGB image is converted to HSV color space and the initializing window is selected by the CAMShift algorithm. This is of help in calculating the backward projection of the color histogram and target position. These calculated values are used as the initial values for the consecutive frame and target tracking is achieved by repeating the process above. The CAMShift algorithm searches the moving targets quickly by adapting to the color feature. The effectiveness of motion detection is improved by combining the background and frame subtraction method.

Figure 2 illustrates framework of the CAMShift algorithm for target detection and tracking, wherefrom the sequential video frames that is, the odd and even frames are identified by the process of frame subtraction and background subtraction. The corresponding results are combined for ROI extraction, for which the histogram and back projection is applied and followed by a mean shift searching for target tracking by centroid movement.

5. Experimentations and Analysis

A simulation was carried out, and the results obtained are considered and analyzed in three directions:

1. The reliability of the proposed algorithm is verified with KTH Human video datasets;
2. The effect of the algorithm on detecting and tracking humans in CAVIR dataset is verified;

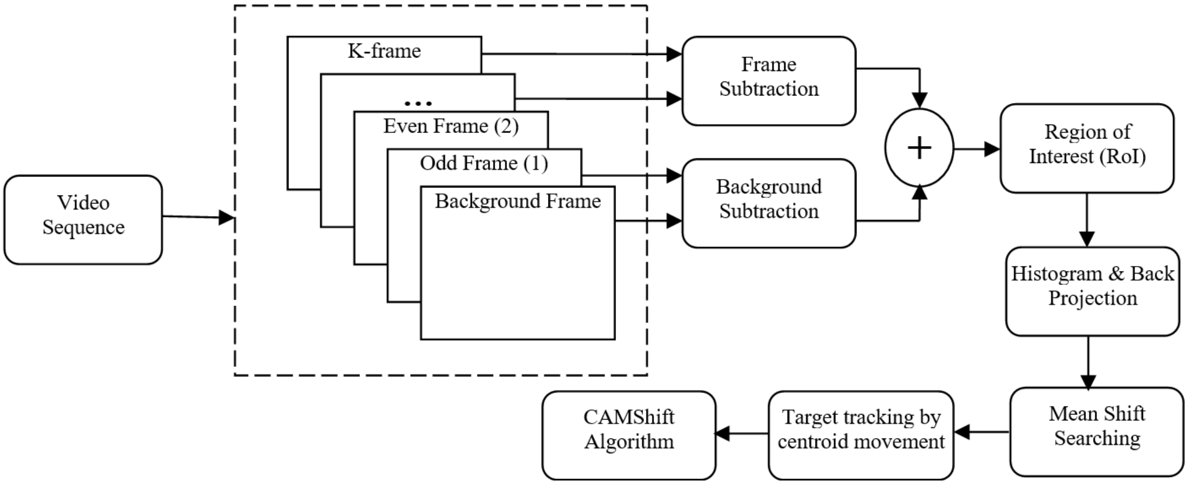

Figure 2. The framework of CAMShift algorithm for target detection and tracking

3. The advantages of the proposed algorithm are verified by comparing it with the CAMShift algorithm.

MATLAB-R2019a software is used for implementing the human detection and tracking system. The performance of different algorithms was evaluated by considering the average tracking error. The average tracking error e_i is calculated by using equation (6):

$$e_i = \frac{1}{N}\sum_{i=1}^{N}\left|T_i - C_i\right| \tag{6}$$

where T_i is the center coordinate and C_i is the actual coordinate of the initial frame.

5.1 Verifying the Reliability of the Proposed Algorithm with the KTH Human Video Dataset

This video dataset consists of 6 types of human actions with 2391 video sequences taken in the homogeneous background with a frame rate of 25 fps using a static camera. The proposed method combines the background subtraction and frame subtraction method. The proposed algorithm is compared with the frame subtraction method of KTH Human video dataset, which consists of holes and many noise points. The absolute difference between successive frames can be easily highlighted. These highlighted regions are considered as the motion region of the moving target. The holes that exist in the frame subtraction method, are due to the incomplete extraction of background information.

The areas with changes in brightness are often detected as false positives and also the frame rate is sometimes too fast for this method. Our proposed method can efficiently model the moving background and outperforming results were obtained.

Figure 3 (a) above shows the original KTH human walking video frame Figure 3 (b) shows the output of the CAMShift algorithm and Figure 3 (c) illustrates the output of the proposed algorithm. In the figure above, it can be noticed that the output obtained from the CAMShift algorithm occupies a large window size in identifying humans, whereas the proposed output identifies the moving human target exactly.

5.2 Validation of Target Tracking Results for the CAVIR Dataset

In occluded environments, the effectiveness of the proposed algorithm is achieved by tracking the multiple moving targets. When a human is moving or hidden in an interacting group, the concealed human can also be tracked. In an outdoor environment, the proposed method shows outperforming results. The anti-occlusion ability of the proposed algorithm is tested when the moving target is deformed in various environmental lighting conditions. The actual position of the moving object is accurately calibrated by the algorithm. Even when the moving targets get occluded it can be indicated exactly and the stability of tracking is ensured.

5.3 Comparison of Simulation and CAMShift Algorithm Results

In the test video, the moving target and the shadow color appear similar to each other. Here,

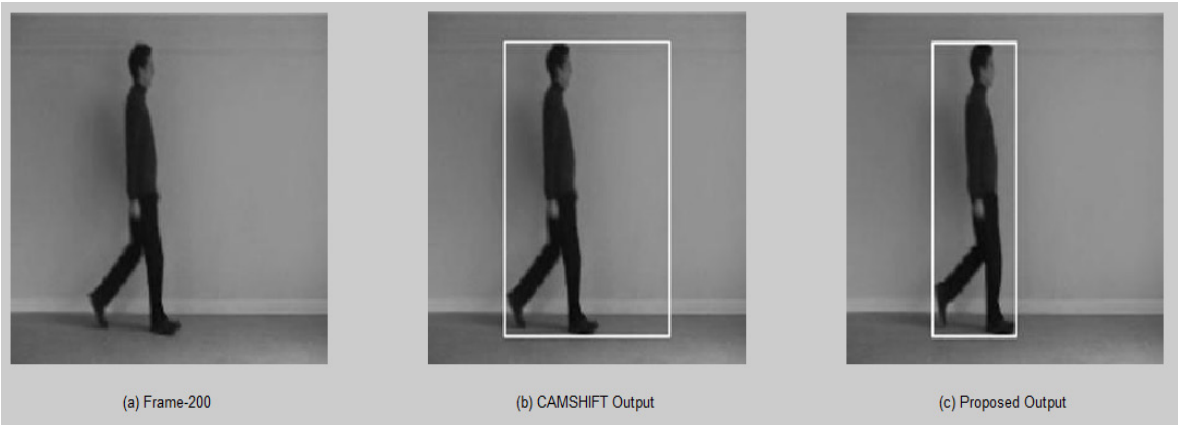

Figure 3. The reliability of the proposed algorithm is verified with the KTH Human video dataset

the pedestrians moving at constant speed are selected as the target. This video has background interference. The Euclidean distance between the real position and the tracking results denotes center positioning error. The best result is achieved when there is an increase in the Euclidean distance value. When the difference between actual coordinates of the target and tracking coordinates is small, the scale control performance and control performance is stable and low for CAMShift algorithm. The quantitative analysis of the existing and proposed algorithm based on Center Positioning Error (CPE) and Distance Accuracy (DA) is carried out where CPE denotes the Euclidean distance between the center of the detected position of a target and the actual target and DA is defined as evaluation index of distance accuracy.

Table 1 shows the comparison results for the CAMShift algorithm and the proposed algorithm for three parameters such as Center Positioning Error, Distance Accuracy, and average target tracking error. In the table above, the results obtained show a major difference in the values of the three parameters analyzed and prove that the proposed method gives outperforming results. It can be concluded from the analysis that target tracking is difficult when the grayscale information is used and similarly when the background color matches with the target color, the target tracking is very difficult in the CAMShift algorithm.

Table 1. Comparison results for the simulation and CAMShift algorithm

	CPE	DA	e_i
CAMSHIFT algorithm	8.07	78.53	100
Proposed algorithm	6.96	91.09	32

From Table 1 it can be noticed that the proposed algorithm features a higher tracking accuracy and lower average target tracking error compared to the CAMShift algorithm. Figure 4 shows the computational time for the proposed algorithm and CAMShift algorithm. The specification of the system is given as follows.

System Specification:

Processor: Intel i7, 8th Gen quad-core

Clock Speed: 1.8 GHz

RAM: 16 GB

Storage: 500 GB SSD

GPU: Nvidia MX

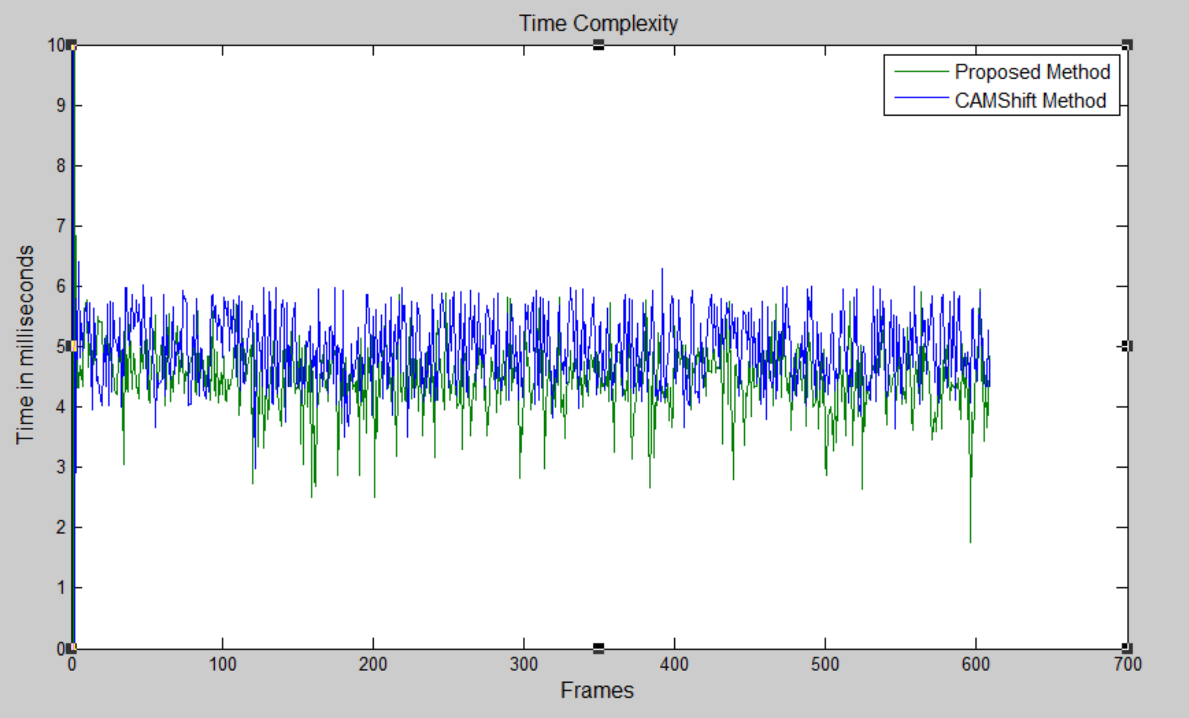

Figure 4. Comparing the time complexity for proposed algorithm and CAMShift algorithm

Time complexity is an important parameter used for live video computing. Figure 4 shows the time needed for processing video frames. The plot illustrates the total number of frames and time in milliseconds for the proposed method and the CAMShift algorithm. This graph shows that the time needed for processing the video frames in case of the proposed method is comparatively minimum.

Figures 5 (a), (b), (c) show the sequential frames in the original video in which frames 95, 114, and 468 are considered for comparing the CAMShift algorithm results with the output of the proposed algorithm. Figures 5 (d), (e), (f) illustrate the results obtained by using the CAMShift algorithm, and Figures 5 (g), (h), (i) show results obtained by using the proposed algorithm, where the major difference was identified in finding the exact target in Figures 5 (f) and (i).

5.4 Comparing the Proposed Results with those of the State-of-the-art Algorithms

The proposed results are compared with those of the state-of-the-art algorithms YOLO and Fast RCNN. The performance parameters, True Positive, True Negative, False Positive, False Negative, Precision, Accuracy, and Recall are calculated in three different environmental conditions that is day bright vision, cloudy vision, and night vision. The F1 score parameter is the balance between recall and precision. Table 2 includes the performance metrics for day bright vision in which case the F1 score of the proposed algorithm shows very mild deviations in the third decimal point and it is negligible. Table 3 includes the performance metrics for cloudy vision in which case the F1 score shows considerable variations. The performance of the proposed approach for

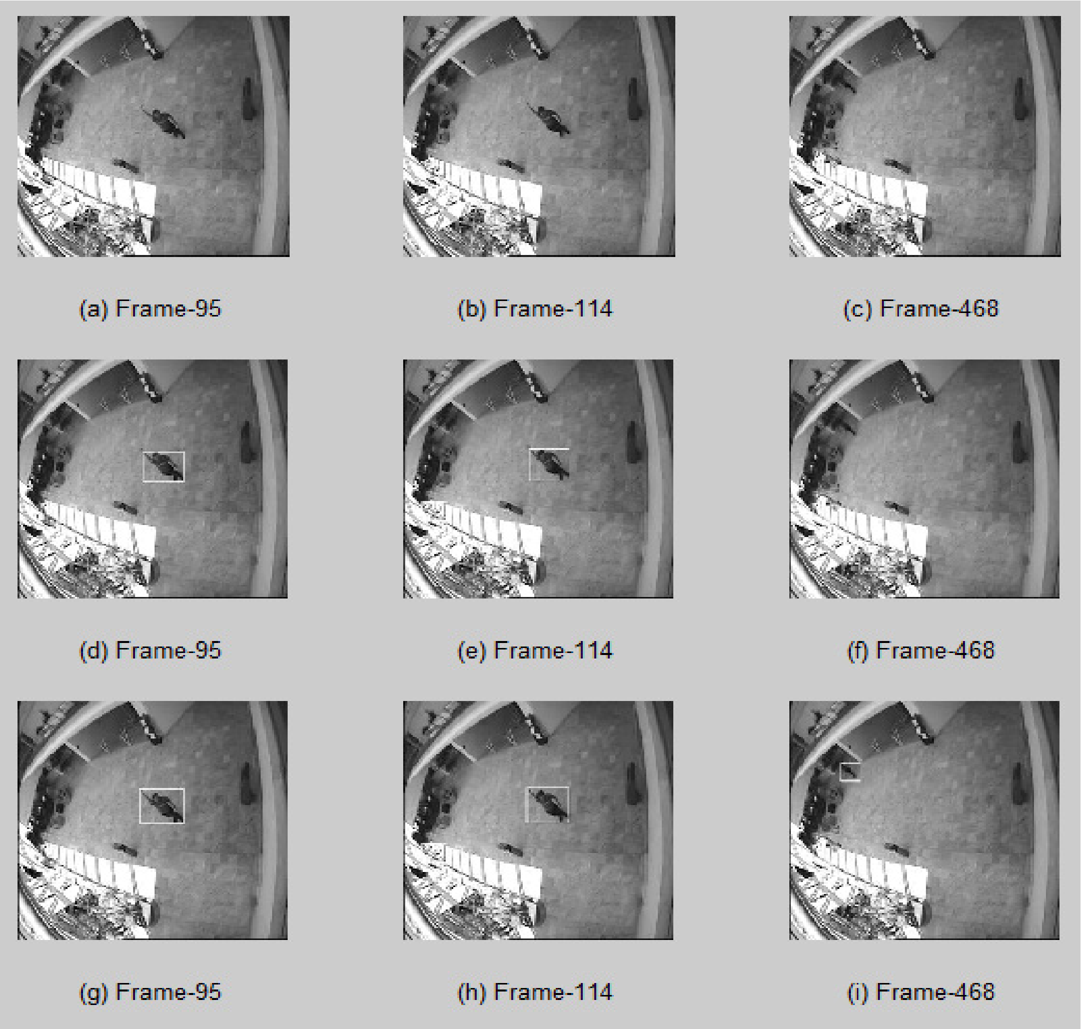

Figure 5. Comparison of simulation and CAMShift algorithm results

Table 2. Performance Metrics for Day Bright Vision

	TP	TN	FP	FN	Precision	Accuracy	Recall	F1-Score
YOLO	2369	4918	161	161	0.936	0.957	0.936	0.9360
Fast RCNN	2346	4965	114	184	0.953	0.960	0.927	0.9398
Proposed approach	2413	4853	206	136	0.921	0.955	0.946	0.9333

Table 3. Performance Metrics for Cloudy Vision

	TP	TN	FP	FN	Precision	Accuracy	Recall	F1-Score
YOLO	1456	2954	58	50	0.961	0.916	0.966	0.9635
Fast RCNN	1430	2980	32	76	0.978	0.906	0.949	0.9633
Proposed approach	1480	2950	62	26	0.959	0.920	0.982	0.9704

cloudy vision is better than that of the YOLO and Fast RCNN algorithms.

Table 4 shows the performance metrics for night vision in which case the F1 score shows considerable variations and it is taken into account. The accuracy parameter values for the YOLO and Fast RCNN algorithms are very close and the accuracy values for the Fast RCNN algorithm and the proposed approach are identical and equal to 0.913. In this case, the F1 score value is considered and the results show that the performance of the proposed approach for cloudy vision is better than that of the Fast RCNN algorithm and lower than that of the YOLO algorithm.

Figure 6 shows the F1 score performance of the proposed approach vs. that of the state-of-the-art algorithms YOLO and Fast RCNN in day bright vision, cloudy vision, and night vision conditions. From the plot, it can be seen that the values of all three algorithms show a mild variation and the proposed approach shows the best performance in cloudy vision. For training

Table 4. Performance Metrics for Night Vision

	TP	TN	FP	FN	Precision	Accuracy	Recall	F1-Score
YOLO	110	232	16	14	0.873	0.919	0.887	0.8799
Fast RCNN	104	236	14	18	0.867	0.913	0.852	0.8594
Proposed approach	108	232	16	16	0.870	0.913	0.870	0.8700

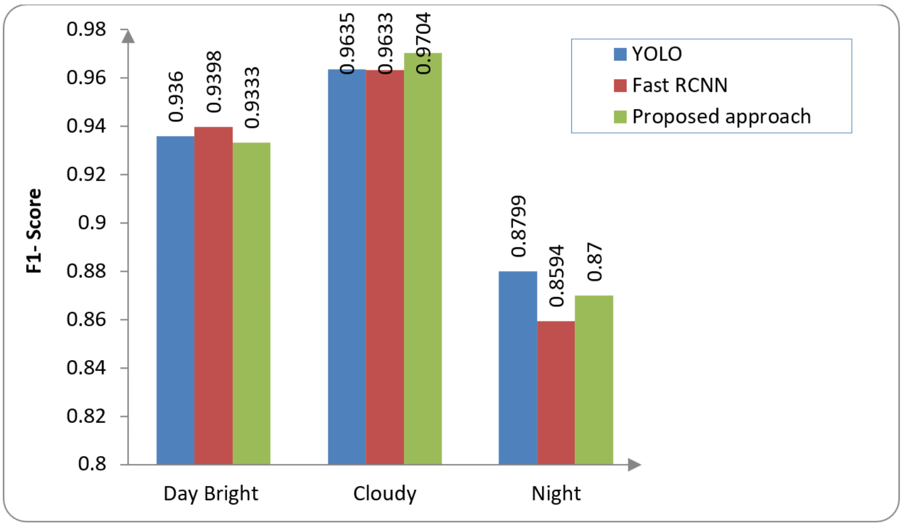

Figure 6. F1 score performance for the proposed approach vs. that of the state-of-the-art algorithms

and testing of YOLO architecture and Fast RCNN architectures, a powerful GPU is required. This results in increased computational costs and high time complexity.

6. Conclusion

The target features extracted from the test video sequence were analyzed for detecting and tracking the target. The information on the foreground target area is selected by applying the required matching algorithm. The obtained features are matched with the path followed by the target with the purpose of obtaining a high target tracking accuracy. In this article, the combined frame subtraction and background subtraction methods support the CAMShift algorithm in finding the exact target position. The various drawbacks of applying CAMShift algorithm such as an increased tracking error and Center Positioning Error (CPE), a decreased Distance Accuracy (DA), and an increased time complexity are overcome by combining frame subtraction and background subtraction methods along with CAMShift algorithm. The simulation results show that the proposed algorithm tracks fast-moving objects in the video frames efficiently and features a high robustness. The F1-score performance comparison plot for all the three analyzed algorithms (Proposed algorithm, YOLO, Fast RCNN) shows considerable variations. If one considers implementation complexity and computation time of YOLO and Fast RCNN algorithms, the proposed approach yields better results and it has also been proved that the proposed algorithm can be widely used in tracking moving objects in the field of video surveillance.

REFERENCES

Ahmed, S. A., Dogra, D. P., Kar, S. & Roy, P. P. (2018). Trajectory-based surveillance analysis: A survey, *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(7), 1985-1997. DOI: 10.1109/TCSVT.2018.2857489.

Akli, B. M., Abdelkrim, N., Fatima, H. & Fethi, D. (2019). November. Moving Objects Detection and Tracking with Camera Motion Compensation. In *International Conference on Electrical Engineering and Control Applications* (pp. 1193-1210). Springer, Singapore. DOI: 10.1007/978-981-15-6403-1_84

Algabri, R. & Choi, M. T. (2020). Deep-learning-based indoor human following of mobile robot using color feature, *Sensors*, *20*(9), p. 2699. DOI: 10.3390/s20092699

Bay, H., Tuytelaars, T. & Van Gool, L. (2006). Surf: Speeded up robust features. In *European Conference on Computer Vision* (pp. 404-417). Springer, Berlin, Heidelberg. DOI: 10.1007/11744023_32

Borji, A., Cheng, M. M., Hou, Q., Jiang, H. & Li, J. (2019). Salient object detection: A survey, *Computational Visual Media*, *5*(2), 117-150. DOI: 10.1007/s41095-019-0149-9

Bouwmans, T., Silva, C., Marghes, C., Zitouni, M. S., Bhaskar, H. & Frelicot, C. (2018). On the role and the importance of features for background modeling and foreground detection, *Computer Science Review*, *28*, 26-91. DOI: 10.1016/j.cosrev.2018.01.004

Cao, S., Wang, X. & Xiang, K. (2017). Visual object tracking based on Motion-Adaptive Particle Filter under complex dynamics, *EURASIP Journal on Image and Video Processing*, *2017*(1), 1-21. DOI: 10.1186/s13640-017-0223-0

Chen, Y., Wang, J., Xia, R., Zhang, Q., Cao, Z. & Yang, K. (2019). The visual object tracking algorithm research based on adaptive combination kernel, *Journal of Ambient Intelligence and Humanized Computing*, *10*(12), 4855-4867. DOI: 10.1007/s12652-018-01171-4

Cheok, M. J., Omar, Z. & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques, *International Journal of Machine Learning and Cybernetics*, *10*(1), 131-153. DOI: 10.1007/s13042-017-0705-5

Chrysos, G. G., Antonakos, E., Snape, P., Asthana, A. & Zafeiriou, S. (2018). A comprehensive performance evaluation of deformable face tracking “in-the-wild”, *International Journal of Computer Vision*, *126*(2), 198-232. DOI: 10.1007/s11263-017-0999-5

Dakua, S. P., Abinahed, J., Zakaria, A., Balakrishnan, S., Younes, G., Navkar, N., Al-Ansari, A., Zhai, X., Bensaali, F. & Amira, A. (2019). Moving object tracking in clinical scenarios: application to cardiac surgery and cerebral aneurysm clipping, *International Journal of Computer Assisted Radiology and Surgery, 14*(12), 2165-2176. DOI: 10.1007/s11548-019-02030-z

Dou, J., Qin, Q. & Tu, Z. (2019). Background subtraction based on deep convolutional neural networks features, *Multimedia Tools and Applications*, *78*(11), 14549-14571. DOI: 10.1007/s11042-018-6854-z

Guerrero-Ibáñez, J., Zeadally, S. & Contreras-Castillo, J. (2018). Sensor technologies for intelligent transportation systems, *Sensors, 18*(4), p. 1212. DOI: 10.3390/s18041212

Kim, C., Lee, J., Han, T. & Kim, Y. M. (2018). A hybrid framework combining background subtraction and deep neural networks for rapid person detection, *Journal of Big Data*, *5*(1), 1-24. DOI: 10.1186/s40537-018-0131-x

Manikandaprabu, N. & Vijayachitra, S. (2019). Adaptive Visual Tracking for Human Motion Detection, *International Journal of Image Processing and Pattern Recognition*, *5*(1), 1-5. DOI: 10.37628/ijoippr.v5i1.484

Naji, S., Jalab, H. A. & Kareem, S. A. (2019). A survey on skin detection in colored images, *Artificial Intelligence Review*, *52*(2), 1041-1087. DOI: 10.1007/s10462-018-9664-9

NaNa, Z. & Jin, Z. (2018). Optimization of face tracking based on KCF and Camshift, *Procedia Computer Science*, *131*, 158-166. DOI: 10.1016/j.procs.2018.04.199

Rahmaniar, W., Wang, W. J. & Chen, H. C. (2019). Real-time detection and recognition of multiple moving objects for aerial surveillance, *Electronics*, *8*(12), p. 1373. DOI: 10.3390/electronics8121373

Ranjan, R., Patel, V. M. & Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(1), 121-135. DOI: 10.1109/TPAMI.2017.2781233

Ray, K. S. & Chakraborty, S. (2017a). An efficient approach for object detection and tracking of objects in a video with variable background, *arXiv preprint* [Online]. Available at: <http://arxiv.org/abs/1706.02672>.

Ray, K. S. & Chakraborty, S. (2019). Object detection by spatio-temporal analysis and tracking of the detected objects in a video with variable background, *Journal of Visual Communication and Image Representation*, *58*, 662-674. DOI: 10.1016/j.jvcir.2018.12.002

Ray, K. S., Dutta, S. & Chakraborty, A. (2017b). Detection, Recognition and Tracking of Moving Objects from Real-time Video via SP Theory of Intelligence and Species Inspired PSO, *arXiv preprint* [Online]. Available at: <http://arxiv.org/abs/1704.07312>.

Redmon, J. & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7263-7271). DOI: 10.1109/CVPR.2017.690

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779-788). DOI: 10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R. & Sun, J. (2016). Faster R-CNN: towards real-time object detection with region proposal networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137-1149. DOI: 10.1109/TPAMI.2016.2577031

Sliti, O., Hamam, H. & Amiri, H. (2018). CLBP for scale and orientation adaptive mean shift tracking, *Journal of King Saud University-Computer and Information Sciences*, *30*(3), 416-429. DOI: 10.1016/j.jksuci.2017.05.003

Sultana, M., Mahmood, A. & Jung, S. K. (2020). Unsupervised Moving Object Detection in Complex Scenes Using Adversarial Regularizations, *IEEE Transactions on Multimedia*. DOI: 10.1109/TMM.2020.3006419

Thabet, E., Khalid, F., Sulaiman, P. S. & Yaakob, R. (2021). Algorithm of local features fusion and modified covariance-matrix technique for hand motion position estimation and hand gesture trajectory tracking approach, *Multimedia Tools and Applications*, *80*(4), 5287-5318. DOI: 10.1007/s11042-020-09903-5

Xia, T., Fan, H., Yu, S., Zhang, L. & Wen, J. (2018). An improved multi-target tracking algorithm for pedestrian counting, *Journal of Physics: Conference Series*, *1069*(1), p. 012113. IOP Publishing. DOI: 10.1088/1742-6596/1069/1/012113

Zhao, H., Xiang, K., Cao, S. & Wang, X. (2016). Robust visual tracking via CAMShift and structural local sparse appearance model, *Journal of Visual Communication and Image Representation*, *34*, 176-186. DOI: 10.1016/j.jvcir.2015.11.008

Zhou, L. & Zhang, J. (2019). Combined Kalman Filter and Multifeature Fusion Siamese Network for Real-Time Visual Tracking, *Sensors*, *19*(9), p. 2201. DOI: 10.3390/s19092201