Journal of Computing and Applied Informatics (JoCAI) Vol. 5, No. 2, 2021 | 102 – 116

DATA SCIENCE
Journal of Computing and Applied Informatics

Implementation of Moving Object Tracker System

Mohanad Abdulhamid[1] and Adam Olalo[2]

[1,2][AL-Hikma University, Iraq, e-mail: moh1hamid@yahoo.com; University of Nairobi, Kenya, e-mail: researcher12018@yahoo.com]

Abstract. The field of computer vision is increasingly becoming an active area of research with tremendous efforts being put towards giving computers the capability of sight. As human beings we are able to see, distinguish between different objects based on their unique features and even trace their movements if they are within our view. For computers to really see they also need to have the capability of identifying different objects and equally track them. This paper focuses on that aspect of identifying objects which the user chooses; the object chosen is differentiated from other objects by comparison of pixel characteristics. The chosen object is then to be tracked with a bounding box for ease of identification of the object's location. A real time video feed captured by a web camera is to be utilized and it's from this environment visible within the camera view that an object is to be selected and tracked. The scope of this paper mainly focuses on the development of a software application that will achieve real time object tracking. The software module will allow the user to identify the object of interest someone wishes to track, while the algorithm employed will enable noise and size filtering for ease of tracking of the object.

Keyword: Implementation, Moving Object Tracker, Computer Vision

Received 28 June 2021 | Accepted 29 September 2021

1 Introduction

The use of video cameras is becoming common in applications such as traffic monitoring, ATM security cameras, surveillance in shopping malls, banking halls and even police surveillance. The video displayed by cameras represents a sequence of images being streamed as successive frames.

Tracking in image processing refers to finding the location of an object from the video being processed and following it up in each successive frame being received. Various algorithms that enable tracking by video cameras are used, but various processes exist that cut across most of these methods. They include processes such as gray scaling, thresholding, size filtering, labeling and noise filtering.

*Corresponding author at: Al Hikma University, Iraq

E-mail address: moh1hamid@yahoo.com

ISSN: 2580-6769 | e-ISSN: 2580-829X | https://doi.org/10.32734/jocai.v5.i2-6450
Journal Homepage: https://jocai.usu.ac.id

For a computer vision system to function effectively as the equivalent biological system, it should be able to cope with static and dynamic background environments, moving and changing objects, changing viewpoints and even changes in illumination.

With the exception of illumination changes and scintillating motion of features in the environment such as water bodies and leaves; for most rigid bodies, changes are usually due to object or camera motion. Moving object detection can be classified as follows:

(A)-Based on the camera state

1-Stationary camera moving object case.

2-Moving camera moving object case.

(B)-Based on the number of objects being tracked

1-Single object tracking in real time.

2-Multiple objects tracking in real time.

(C)-Based on the number of cameras being used to track

1-Single camera tracking.

2-Multiple camera tracking.

Some works related to our topic can be found in the literatures [1]-[7].

2 Implementation

In this paper, the object tracking system is developed and implemented using Matlab as illustrated in the following sections.

2.1 Main Flow Chart

The proposed approach can be illustrated with a flowchart as shown in Figure 1. It starts with image acquisition of the live video feed by the web camera after which the user initiates the object of interest which is identified by various selection algorithms. If the start tracking button is pressed, a flag value which indicates whether the button is pressed or not is checked. If it is pressed, motion is detected in the video and the motion of the specific object of interest localized thereby achieving tracking of the object; otherwise, tracking is not initiated unless the button is pressed.

On the other hand, if the stop tracking button has not been pressed tracking continues infinitely as long as the object is within the camera view, but once it's pressed tracking stops and the user can either exit the application or initiate a different object to track. Similarly, there also exists a flag value that keeps track of the state of the stop tracking button (whether it's pressed or not). The next sections explain the processes of the main program flowchart in Figure 1 in detail.

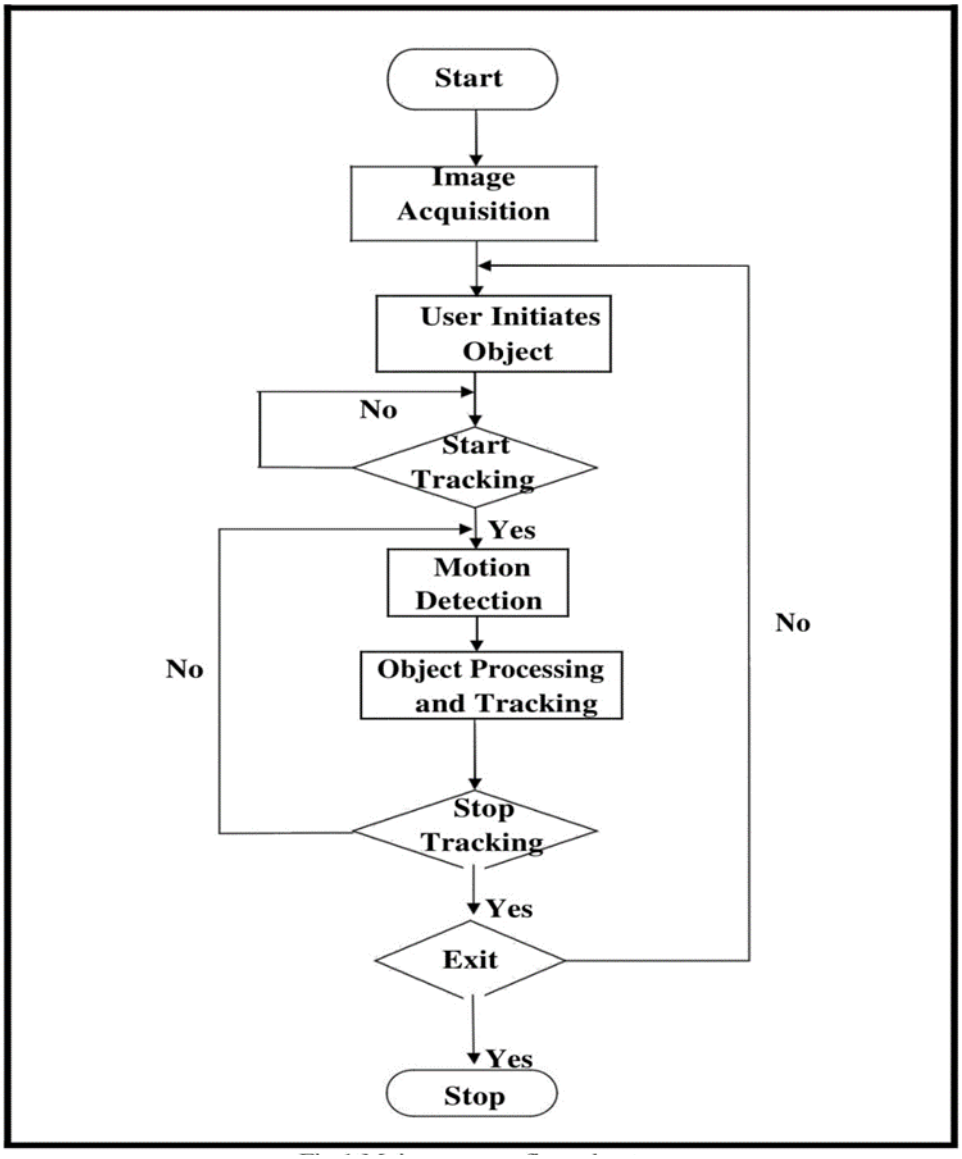

Figure 1 Main Program Flow Chart

2.1.1 Image Acquisition Process

Figure 2 shows the flow chart for the image acquisition process. This process launches the graphical user interface (GUI) handle onto which the video will displayed, it also contains the command buttons for initiating tracking, stopping and exiting. In addition, the live video acquired by the web camera is also previewed at the GUI handle at this juncture.

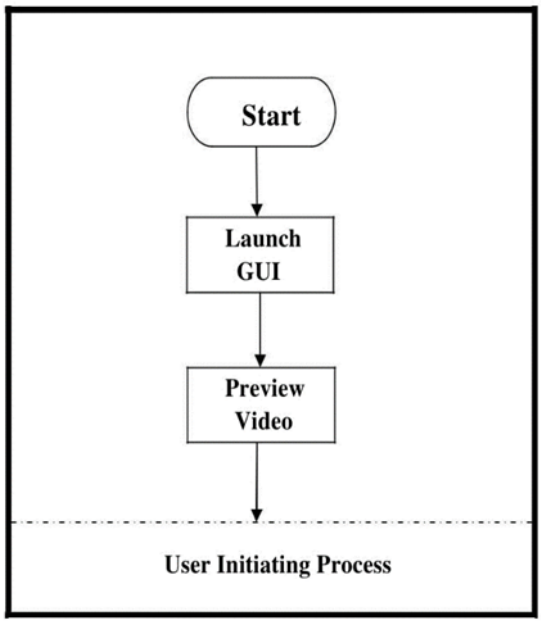

Figure 2. Image Acquisition Process

2.1.2 User Initiating Process

Figure 3 shows the detailed flow chart for the user initiating process. In this process the user first selects the object someone wishes to track with a resizable rectangle, the selected object is then modeled by its unique properties (colour and pixel intensity). It is these unique model properties of the object that aid in the object's segmentation and its subsequent tracking in the midst of other objects.

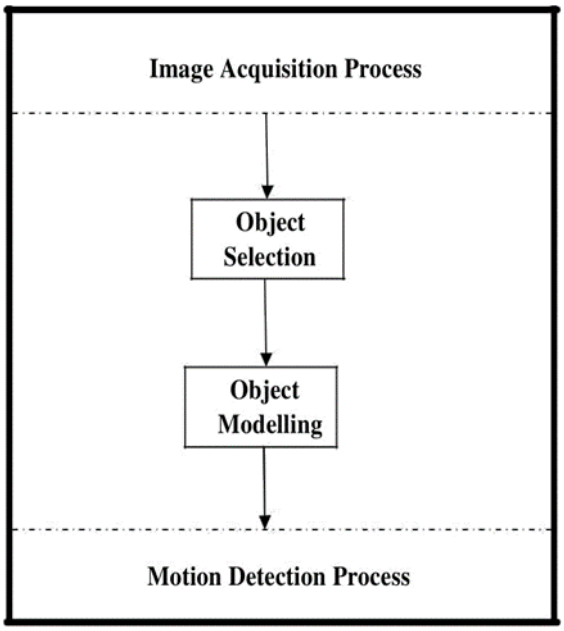

Figure 3. User Initiating Process

2.1.3 Motion Detection Process

Figure 4 illustrates the detailed motion detection process flow chart. Before motion detection commences, first there is the initialization of the tracking process which consists of the video input object capturing the web camera video feed as per the set frame grabber properties. The frame grabber sets the number of frames captured by the video input object per second and also enables capturing the snapshot of the current frame.

From the current frame snapshot, the object of interest is segmented according to the user-initiated object properties (colour and pixel intensity) which were used in its modeling. After that motion is extracted using background subtraction technique and possible regions of motion of the object localized within image sequences as successive frames are grabbed. It is from these successive frames that the motion of the object of interest will be segmented and tracked accordingly.

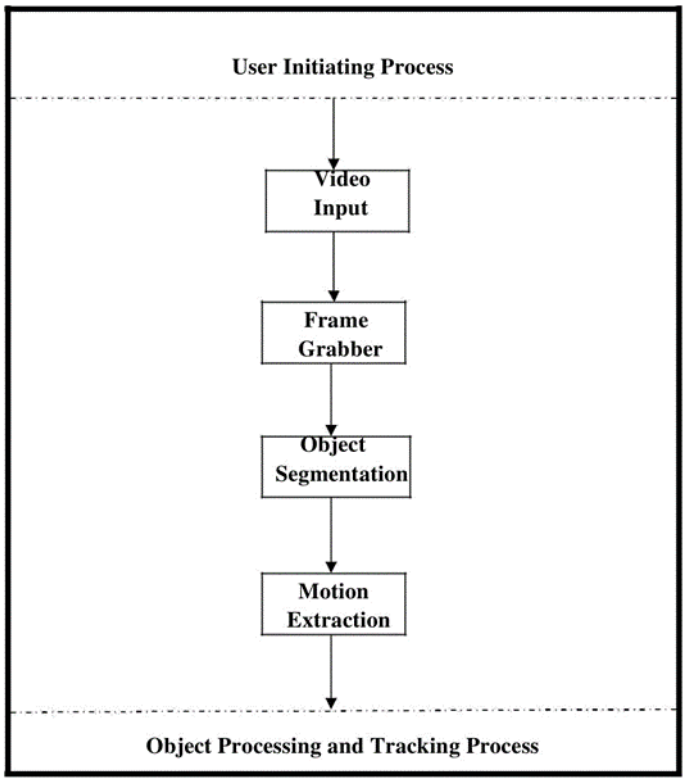

Figure 4. Motion Detection Process

2.1.4 Object Processing and Tracking Process

In this process (Figure 5), the moving region with the probable location of the object of interest is gray thresholded and then binary masked so as to separate the foreground from the background; our object of interest will be represented by the foreground (or binary mask '1', with the background marked as binary mask '0').

To get a precise mask representing the object, noise and size filtering is carried out so as remove pixels that can be mistaken to represent the object. After that object labeling is carried out from the moving region and it is this labeled object that will be tracked by a bounding box.

After tracking is stopped by the user, the video input object is deleted, and memory cleared so as to free up memory resources of the system which were being utilized in tracking of the object.

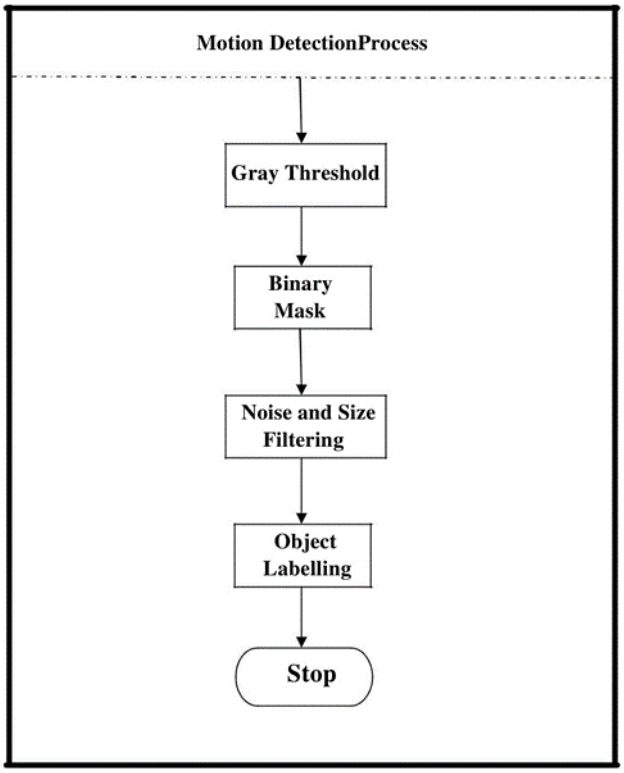

Figure 5. Object Processing and Tracking Process

2.2 Program Algorithm

The program implementation stages can be summarized by the following algorithms:

- Using Matlab's video input object the installed adapter of the web camera is called by the program.
- Matlab's preview function utilizes the installed adapter of the web camera to acquire its video feed and displays it in real time at the already launched GUI.
- User initiates object selection.
- An algorithm is then employed that grabs a certain number of frames per second from image sequences being received.
- A model of the foreground is then computed from the current frame based on the color and pixel intensity values of the user-initiated object.
- Background subtraction algorithm is then carried out so as to segment the foreground region which depicts the probable object of interest.

- Morphological dilation is then carried out so as to close the small gap regions of the segmented foreground.
- Further segmentation of the object from the foreground region is attained by employing noise and size filtering algorithms.
- Objects are then labeled based on the number of connected components in the filtered foreground region by employing contour finder techniques. From these labeled objects our object of interest is identified and uniquely tracked.

3 Experimental Results

This section illustrates the results of the program when tested with different video sequences provided. The program algorithm is as per developed using the Matlab's environment and the video sequences represent the real time data acquired using a low-resolution webcam with an image size of 320*240 pixels.

In this analysis two video sequences are taken into consideration: one for two different coloured objects and another for similar objects. The results are analysed stepwise from object segmentation, background subtraction, noise and size filtering, object labelling, and the final results as perceived by the user.

Figure 6 and Figure 7 give an illustration of the two captured video sequences as perceived by the web camera, while Figure 8 and Figure 9 indicate the user selected regions in the respective video sequences.

Figure 6. Snapshot of the 1st Video Sequence

Figure 7. Snapshot of the 2nd Video Sequence

Figure 8. User Selected Object of 1st Sequence

Figure 9. User Selected Object of 2nd Sequence

3.1 Results from Object Segmentation

Figure 10 and Figure 11 illustrate the binary image representation of the first frame for both video sequences; the white regions represent the possible locations of the object as per the set color constraints used in segmentation of the object.

Figure 10. Object Segmentation of 1st Sequence

Figure 11. Object Segmentation of 2nd Sequence

3.2 Results from Background Subtraction

Background subtraction detects motion present globally thereby distinguishing the moving object from a static background. Background subtraction alone does not show a clear contrast between the object and the static background in some cases because the camera reads an image by its pixel values. Hence in this program gray thresholding is applied prior to the background subtraction process leading to a static background that is enhanced and more apparent as shown in Figure 12 and Figure 13.

Figure 12. Background Subtraction of 1st Sequence

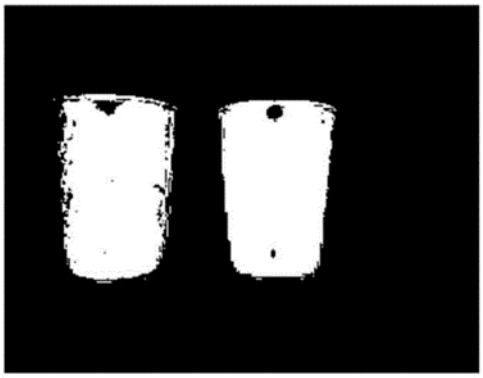

Figure 13. Background Subtraction of 2nd Sequence

3.3 Results from Noise and Size Filtering

This process has the effect of removing noise and small sized pixels which are less than 300px that may be mistakenly taken to represent the object. Thereby leading to a sharp image that is free from noise with a clear distinction of the possible object location as per the set colour range constraints. This effect is illustrated in the video sequences shown in Figure 14 and Figure 15, with Figure 15 showing two possible object detections since the two objects are of similar characteristics.

Figure 14. Filtering of 1st Video Sequence

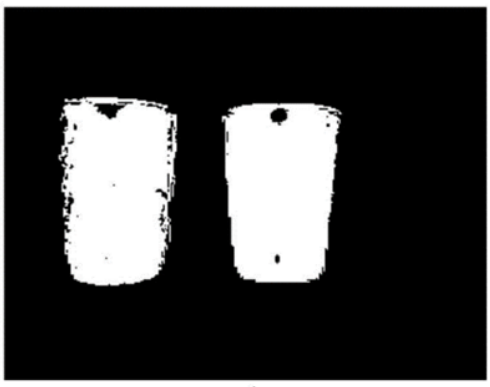

Figure 15. Filtering of 2nd Video Sequence

3.4 Results from Object Labeling

In this procedure, all the connected components are labeled, and every connected component is uniquely identified as a different object by the program. From the visual point of view there is no difference between the results of object labeling and those of filtering, but actually the detected objects are labeled at this instance at the backend of the program.

From the 1st video sequence only one object was detected as shown in Figure 16, which was thus labeled as number 1. From the second video sequence two objects were detected as shown in Figure 17, the one on the left was labeled as number 1 while the one on the right was labeled as number two.

Based on the assumption that the selected object would not have moved significantly from the instance of object selection to when the object is detected in the next captured frame. A relative position of the object is computed and it's on this positional basis that the rightful selected object is tracked in the second video sequence.

Figure 16. Object Labeling of 1st Sequence

Figure 17. Object Labeling of 2nd Sequence

3.5 Final Results from the User Point of View

When a moving object specified by the user is being detected a blue boundary box is drawn surrounding the object, as the program's algorithm is implemented between successive frames the bounding box moves with the object.

Figure 18 to Figure 21 illustrate the motion of the chosen object in various frames as perceived by the program from web camera feed of the first video sequence.

Figure 22 to Figure 25 also illustrate the motion of the chosen object as perceived by the program from the web camera feed of the second video sequence.

From both video sequences, the selected object was successfully detected and tracked by a blue bounding box as it moved within the camera view.

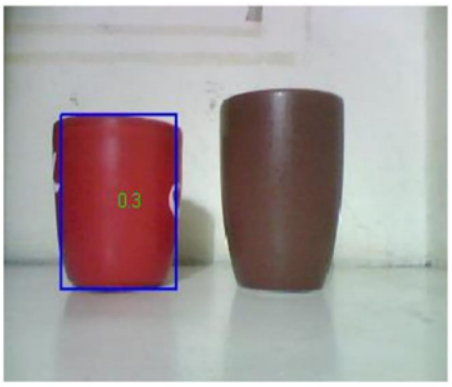

Figure 18. 1st Frame Result of 1st Video Sequence

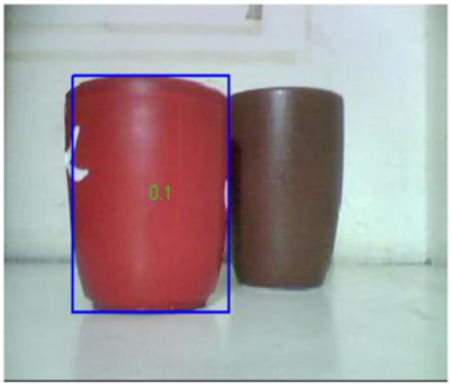

Figure 19. 5th Frame Result of 1st Video Sequence

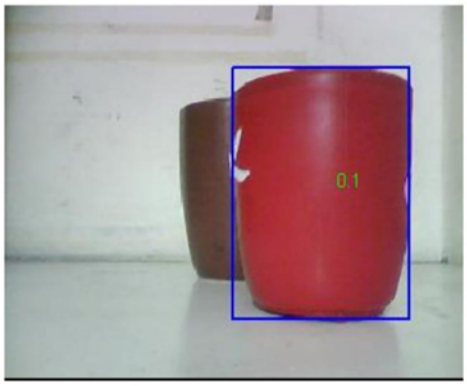

Figure 20. 20th Frame of 1st Video Sequence

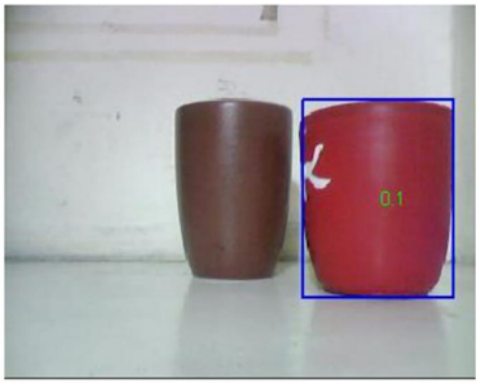

Figure 21. 45th Frame of 1st Video Sequence

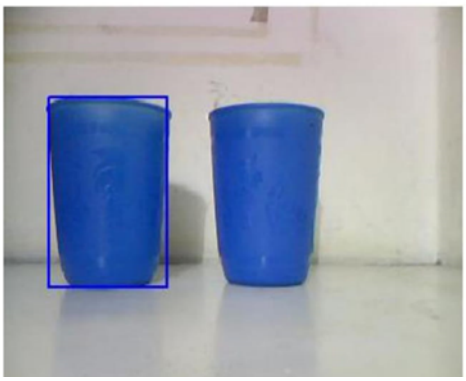

Figure 22. 1st Frame of 2nd Video Sequence

Figure 23. 5th Frame of 2nd Video Sequence

Figure 24. 20th Frame of 2nd Video Sequence

Figure 25. 45th Frame of 2nd Video Sequence

4 Conclusion

The main objective of this paper was to develop a program that demonstrates the tracking of a user selected object of interest that is within a stationary camera field of view. The program achieves this by employing various algorithms for object segmentation, gray thresholding, background subtraction, filtering and object labeling which lead to object localization and its eventual tracking. Though the algorithms employed had a reasonable success rate as far as stationary controlled scenes are concerned; for deployment in real world situations such as video surveillance applications, guiding of autonomous vehicles, automatic target recognition and missile guidance work needs to be done in order to fine tune the algorithms for better tracking performance.

REFERENCES

[1] C. Lee, “Moving object detection at night”, Master thesis in Computer and Microelectronic Systems, Malaysia Technical University, Malaysia, 2007.

[2] Adam Olalo, "Moving object tracker", Graduation Project, University of Nairobi, Kenya, 2011.

[3] S. Parekh, G. Thakore, K. JaJiya, "A survey on object detection and tracking methods", *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, issue 2, 2014.

[4] B. Deori, D. Thounaojam, "A survey on moving object tracking in video", *International Journal on Information Theory*, vol. 3, no. 3, 2014.

[5] P. Panchal, G. Prajapati, S. Patel, H. Shah, J. Nasriwala, "A review on object detection and tracking methods", *International Journal for Research in Emerging Science and Technology*, vol. 2, issue 1, 2015.

[6] S. Balaji, S. Karthikeyan, "A survey on moving object tracking using image processing", *11th International Conference on Intelligent Systems and Control (ISCO)*, 2017.

[7] R. Hatwar, S. Kamble, "A review on moving object detection and tracking methods in video", *International Journal of Pure and Applied Mathematics*, vol. 118, no. 16, 2018.