A New Algorithm Designing for Detection of Moving Objects in Video

Tripty Singh, Ph.D
Assistant Professor
Amrita School of Engineering
Bangalore, India

Sanju S
M.Tech in Embedded Systems
Amrita School of Engineering
Bangalore, India

M.Tech in VLSI Design
Bichu Vijay
Amrita School of Engineering
Bangalore, India

ABSTRACT

Now a day's video motion detection is implemented targeting a wide class of applications such as in autonomous video surveillance strategies for security and vision analysis, detecting human presence in destructed environments, etc. This paper proposes a method for detection of moving objects in highly secured environments where it can be deployed either on a robotic vehicle or at a static permanent position. The robot acquires information about its surroundings through a camera mounted on it in real time. Another objective of this paper is to increase the efficiency of moving objects detection in offline and online video processing mode. In offline mode, an AVI file is read and it is decomposed into frames. Noise removal is done to improve the image quality and segmentation is performed to detect the moving objects in the foreground by separating it from a known static background. Various operations are carried out and the moving object is identified by marking a rectangular box around the detected object in each frame. When a movement is spotted, alarm is activated. The distance between the centroid of the object in the video file is found and thus the velocity of the movement is determined. In online mode, by comparing each and every frame the presence of moving object is checked. When a prowler is detected, the proposed algorithm triggers the alarm. At that instant, the snapshot of the object is generated and from this, the distance to the object is identified. The proposed algorithm is tested with input AVI format video file of 320 x 240 frame size and frame rate 15fps in offline mode. In online mode, video file of 640 x 480 frame size and frame rate 25fps is captured in real time using a webcam. MATLAB is used for the execution of motion detector algorithms. The result is demonstrated in a different real sequence and analysis of algorithm on the basis of its performance is evaluated.

Keywords

Foreground, AVI, Surveillance, Segmentation, Noise Removal

1. INTRODUCTION

Motion detection is usually a software-based monitoring algorithm that will signal, for example in the surveillance camera to begin capturing the event when it detects motion. In object tracking, the object is located and the moving object is followed. One of the fundamental steps in many computer based vision systems for object tracking and motion detection is real-time segmentation of moving regions in the image sequences. Segmentation is done in order to detect the object accurately. Usually cameras are used as input sensors, for recording.

In general, there are several conventional approaches in moving object detection like temporal differencing, background subtraction etc. Temporal differencing is appropriate in dynamic environments. The limitation of this method is that it does a poor job in extracting all relevant feature pixels. In background subtraction, image background and foreground are needed to be separated, processed and analysed and thus bestows entire data features, but it is well sensitive to dynamic scene changes due to lighting and superfluous events. Optical flow, another method is used to detect independently moving objects in the presence of camera motion. This method is computationally complex, and cannot be applied to full frame video streams in real-time. So it is not possible to implement this without specialized hardware. Recent researches put forward several background modelling techniques and algorithms which are based on image differencing, exhibit superior real-time performance and high accuracy.

In both online and offline mode, the selection of threshold values plays a vital role according to the environment of the implementation. The offline mode algorithm is useful in such milieu where security is not much significant. To check the intruder presence after a malfunction is suspected, from a recorded video file this algorithm is used. In offline mode, an AVI file is read and the pixel data are compiled into a single array. Then video pixels are converted to gray scale from RGB scale. Gray scale image background is subtracted from the foreground image. Then the resultant frames are converted into a binary image. Segmentation and the feature extraction from sequences of frames are performed in order to detect the moving object. Then by plotting a rectangular box in each frame, the object is tracked. By checking pixel values in every frame, the position of the object is calculated. The distance between the centroid of objects in the scene is also obtained and thus the velocity of the moving object determined. The online mode algorithm, operating in real time is effective in highly secured regions like intrusion prevention systems that are able to actively prevent/block intrusions and thereby ameliorate security fortification in real time. In this mode, there is an option for starting/stopping video detection. The difference between the two frames is checked continuously. If the difference is greater than a threshold, then a motion is detected and the snapshot of the difference is displayed subsequent to an alarm.

This article is organized as follows. Section 2 gives a brief account of literature survey done and related work done on that topic and brief overview of the system including previous flaws. The proposed model and algorithm of the presented detection scheme is given in Section 3 and in Section 4, implementation of object detection algorithms are explained. In Section 5, results are shown.

2. STATE OF ART

In the algorithm developed by Gottipati Srinivas Babu [1] the background is dynamically updated frame by frame. The shadow of the moving object is identified and removed to calculate the area of the object accurately. This algorithm also identifies even a small object by adjusting the threshold values. Even the smallest, slowest, fastest, of a moving region is detected accurately by selecting the proper threshold value of the objects. Finally this algorithm works for On-line (Real time) and Off-line (Quasi real time) video processing and its computational complexity is low. Here object classifications have to be done by better understanding of human motion not only on the vehicle, including segmentation and tracking of articulated body parts. And improved data logging and retrieval mechanisms to support 24/7 system operations is needed and also better camera control is needed to enable smooth object tracking at high zoom, in case, video is vibrating. For this proper video stabilization algorithm is needed.

An efficient algorithm method for detecting a moving object is depicted in [2] by using a background elimination technique. Initially the frame differences (FD) between frames are computed. The frame differences obtained are then compared with one another which help in identifying the stationary background image. The moving object is then isolated from the background. In the post processing step, the noise and shadow regions present in the moving object are eliminated using a morphological gradient operation that uses median filter without disturbing the object shape. This could be used in real time applications involving multimedia communication systems. The experimental results obtained indicate that the clarity of the image obtained using background elimination technique is much better than using background registration technique. Good segmentation quality is achieved efficiently. This paper also discusses on an application system of traffic surveillance and developed an algorithm to track and count dynamic objects efficiently. Testing on a combination of higher dimensional features with additional constraints in dynamic environment should have resulted in a better outcome.

In [3], general moving object detection and tracking based on vision system using image difference algorithm is done in MATLAB. The result is better compared to other research by using such types of noise removal filter and also such types of structure element for morphological operations. In this velocity as well as acceleration of moving object in a particular area is estimated. In order to get accurate velocity and acceleration, using centroid extra mathematical calculations has to be done.

In the approach explained in [4], it is able to detect motion and extract object information which involves human as object. The algorithm involves modelling of the needed background as a reference model for latter usage in the background subtraction. This is done to produce foreground pixels, the deviation of the current frame from the reference frame. The deviation which represents the moving object within the analysed frame is further processed to localize and extract information. Except on very large cast shadows, sometimes shadow will not get completely removed. This is mainly due to the fact that brightness decreases below the brightness distortion threshold. This problem can be corrected using lower thresholds in the brightness distortion with the drawback of introducing false shadow pixel detection.

For detecting and helping humans in destructed environments, simulated autonomous robot using video processing is illustrated in [5] and [6]. Under conditions where illumination cannot be controlled, moving object detection algorithm based on the shading model method with background updating is used in [7]. This new method is shown to be invariant to significant illumination changes and superior to other techniques when the illumination is allowed to vary. In order to accelerate the execution time of the whole algorithm, in this, processed only certain segments of images where it is expected to find the moving object. It also uses the median value of the observed region, instead of the mean value in calculating the variance because the comparison is faster than addition and division (necessary for calculating the mean value).

3. PROPOSED MODEL

3.1 Offline Mode

The proposed algorithm processes offline video as shown in Fig.1.

After selecting the optimum threshold values, input AVI format video file is read. Then extracting the red, blue, green intensities from each and every frame of the input video is done and it is converted into binary image. Noise removal is done by filter technique in order to get a better image feature. By adjusting the threshold value, Median filter algorithm is used as it is able to remove the salt pepper noise. The median filter is very good in reducing the short noise. Segmentation is done to separate different regions in an image. The feature extraction is performed after segmentation to analyse the regions in image. Then tracking the position of the object is executed to find the distance between the centroid of objects.

3.1.1 Video Acquisition

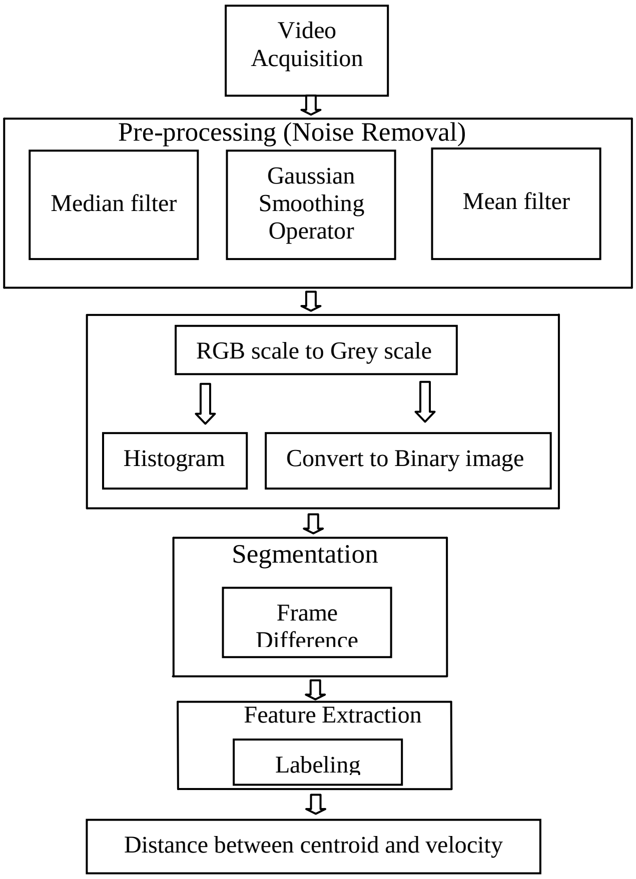


Fig 1: Block Diagram for proposed offline algorithm

The input video format is AVI. AVI stands for audio video interleave. Under the Resource Interchange File Format container format, an AVI file store audio and video data. Video Data is usually stored in AVI files in compressed format with various codec's and parameters. The object tracking is performed with the help of a digital camera. The frame rate of the video used is 15 fps in order to reduce the frame size of the video clip. Less frame size is used because less memory is needed for calculations and position change of an object can be easily tracked from the video stream. The video frames are then divided into several frames.

3.1.2 Pre-processing

Mean Filter is a low pass filter. For noise removal box functionality is used. It starts with scanning the complete image. The mean value is obtained for every pixel. Then in the central element, the pixel values are kept. For example, consider a 3x3 matrix:

$$\begin{bmatrix} 1 & 2 & 9 \\ 4 & 3 & 8 \\ 5 & 6 & 7 \end{bmatrix}$$

Pixel value = $\frac{1}{9}$ * (1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9)

Here the central element is '3'. The value of 3 is substituted by the pixel value. The time for performing a mean filter computation is very less. Thus smoothing is completed by using a median filter.

For blurring the image and noise removal Gaussian smoothing operator is used. Even though Gaussian smoothing is similar to the median filter, here a different function is used to find the pixel value. After scanning the complete image, the pixel value is obtained from the standard deviation of the Gaussian and the value is substituted as central element. The box size varying from 1 to 7 denotes 1x1 to 7x7 elements.

$$\mathrm{G(x,y)} = \left(\frac{1}{2\pi\sigma^2}\right) * \mathrm{e}^{-\frac{\mathrm{X}^2+\mathrm{Y}^2}{2\sigma^2}} \qquad (1)$$

Here x and y represent the element's position in the box.

In the median noise is removed by manipulating the median from the box elements and the value is substituted in the central element. For example, consider a 3x3 matrix:

$$\begin{bmatrix} 1 & 2 & 9 \\ 4 & 3 & 8 \\ 5 & 6 & 7 \end{bmatrix}$$

At first the median filter sorts the elements in the matrix. Then the median value is stored as the central pixel. The median 5 will be assigned to the central element. The box scan is done. Thus the noise is removed, but the execution time is more due to the sorting techniques used. The small pixels are trimmed down in this type of filtering.

3.1.3 Image Conversions

Into the gray scale image:

Gray scale images are images without colour, or achromatic images. The levels of a gray scale range from 0 (black) to 1 (white). Images are converted into gray scale images for reducing the complexity while applying the segmentation operations.

Histogram:

Histograms are functions describing information extracted from the image. In a process called histogram equalization, this information is used to compute a contrast increasing intensity transformation function. The histogram function is defined over all possible intensity levels.

An image histogram is a graphical representation of the number of pixels in an image. Histograms consist of bins. Every bin represents a specific intensity value.

The histogram is computed by examining all pixels in the image. Each pixel is assigned to a bin depending on its pixel intensity. The final value of a bin denotes the number of pixels assigned to it. The histogram of a digital image with L total possible intensity levels in the range [0, G] is defined as the discrete function

$$\mathrm{h}\ (r_k) = n_k \qquad (2)$$

Where $r_k$is the k^{th} intensity level at the interval [0, G] and n_k is the number of pixels in the image whose intensity level isr_k.

Into the binary image:

The frames are again converted into the binary images so that it may help in extracting the images easily and accurately.

3.1.4 Segmentation

There are lots of techniques that are used in segmentation such as Single Gaussian Background Method; Histogram based Segmentation, Frame difference etc. Frame Difference is explained below:

At every pixel position, frame difference calculates the difference between 2 frames. Then the absolute difference is stored. For the calculation, it needs less memory. Let us consider an example; if we take a sequence of frames, the current frame and the future frame are taken into consideration at every calculation. After the calculation the future frame becomes current and the frame that comes in the sequence becomes the future frame.

The frame difference algorithm is used for performing the segmentation process. It requires only a less processing time. In the frame difference algorithm, the separation of sequential frames is done. Initially the input images are read. Then processing starts with considering the current position as first position up to the last position in a loop. After this the difference between the pixel values at current position of sequential images are found. Then the absolute value is obtained and the difference is substituted in the new image at same pixel position (current position).

3.1.5 Feature Extraction

Feature Extraction checks the moving object in sequential frames and a rectangle box is marked around the object using its dimensions. As the algorithm is implemented, depending on the dimensions of the object produced, a rectangular box is generated over the moving object. With the help of box dimensions obtained, the centroid of the box is marked. Thus using centroid the objects can be tracked. Even if there is more than one moving object in the frame as shown in Fig.2, the entire moving objects are tracked.

After performing the feature extraction operations, the area of the moving object is calculated and the moving objects are labelled as red colour in the rectangle box at the output.

3.1.6 Distance between centroid of the objects and velocity

The centroid pixel in each image is checked and from this the distance travelled by an object is obtained. To store position of all the pixel values of the centroid in each frame an array is used. Another array is used to store the distance between earlier and current pixel positions during the implementation. The distance is calculated by using Euclidean distance formula and the value obtained is stored in the array. The velocity of the moving object also can be calculated. The distance centroid travelled to the video frame rate is velocity.

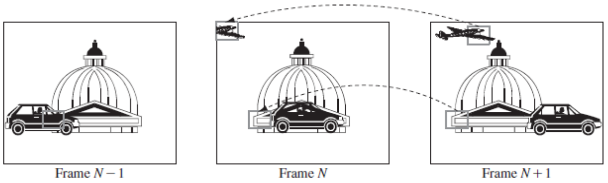


Fig 2: Tracking of moving objects

3.2 Online Mode

The proposed algorithm for online video processing is as shown in the Fig.3.

In this mode, at first video in real time is obtained using webcam/camera. The background noise is adjusted and the sensitivity is calculated. Then the difference between each two successive frames are analysed. If the difference is greater than a threshold, then it is considered as motion detection occurred and the snapshot of the difference image is generated along with an alarm. When the difference is less than the threshold, it is considered to be no motion detected.

3.2.1 Selection of threshold value

The selection of threshold value merely depends upon the object detection environment in which it is supposed to be implemented. This selected threshold value named as sensitivity is compared with the difference in frame to detect the presence of moving object.

3.2.2 Activate/Deactivate Motion Detection

Using a triggering arrangement, the video capturing is activated / deactivated according to the requirement.

3.2.3 Video acquisition

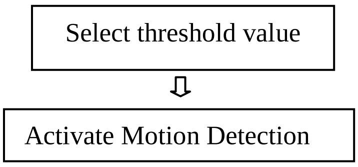
Here the real time video is captured using a webcam/camera. A stationary camera records the field of view. A large monitoring area may lead to too small object images, which will affect the resolution of the object. Here the proposed algorithm is based on a model-free approach.

In Fig.4, the camera axes are indicated by ξ, η, and ζ, with ζ in the direction of the optical axis, ξ in the direction of the scan lines, and η downward to form a right-handed reference frame. Considering the image plane being at unit distance from the projection centre along the optical axis ζ and the image plane coordinates are (x, y) with the x and y axes parallel to ξ and η, respectively.

The camera frame and the world frame are connected by a rotation matrixR^T , where

$$R^T = [\hat{u} \quad \hat{v} \quad \hat{w}] \qquad (3)$$

The video file is of 640x480 frame size and RGB colour space is used. The frame rate is 25 fps. The video is captured only after providing a trigger.



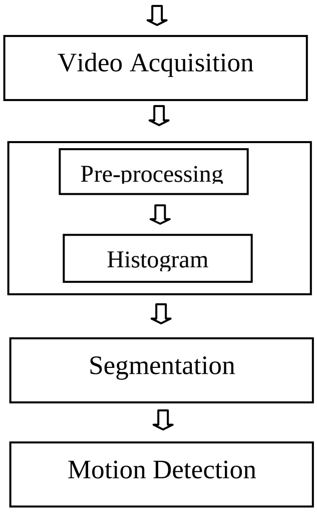


Fig 3: Block Diagram for proposed online algorithm

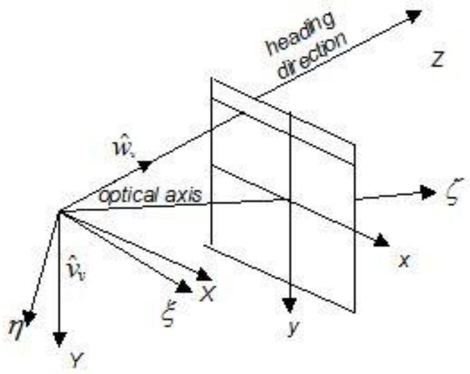


Fig 4: Camera reference frame

3.2.4 Pre-processing

The background noise is reduced using filter technique. Median filtering is done followed by the Gaussian smoothing operator.

Image histogram is an important tool used for the inspection of the images. It allows to spot background and gray value range. Here the histogram of the frame during different instances is used to test the performance of the proposed algorithm.

3.2.5 Segmentation

To separate the moving objects from the whole image in every frame, frame difference algorithm is used. The successive frames are evaluated and the difference is found. When only one object is used in the scene, frame difference algorithm produces better result.

3.2.6 Motion Detection

The difference between the consecutive frames and the sensitivity is compared.

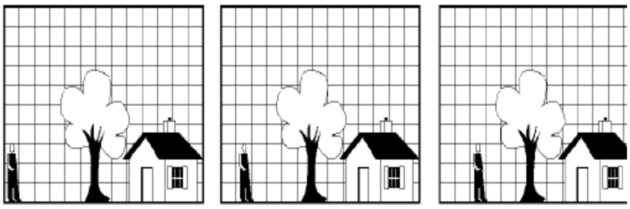


Fig 5: The movement of human in a static frame

If the frame difference is greater than the threshold, an alarm sound is produced. For example, as shown Fig.5 the difference, which is the moving object, is detected. Alarm indicates that a motion is detected and the snapshot of the difference image is also generated. If the frame difference is less than the sensitivity, nothing is changed.

4. IMPLEMENTATION

4.1 Offline mode

In this mode, the algorithms for noise removal, segmentation, feature extraction and distance measurement between two centroid of moving objects and velocity are implemented as mentioned below:

4.1.1 Noise removal

Using the median filter, the noise removal algorithm is implemented. At first each input frames are read. Keeping current position as first position till last position, all the elements are scanned. Here to sort every pixel values, bubble sort technique is used. Then the median is calculated.

Then assign the value to the current position of the input image. The first position becomes second row second column of an image resolution and the last position is (n-1) row (n-1) column of an image resolution. Also the current position varies from first to last position.

4.1.2 Segmentation

The input image is read frame by frame. The difference between the pixel values at current position of sequential images is obtained after defining the current first position and last position. The absolute value is calculated. Then the difference is stored at same pixel position (current position) in the new image.

4.1.3 Feature Extraction:

The image difference is read. For the X and Y resolution, the first/current position and the last position are defined. Then from top and down, the sharp change in intensity of the image is calculated and the values are stored. The bounding box height is the difference between the bottom value and the top value. Again after defining the first and last position, from left and right the sharp change in intensity of the image is determined. Then in an array the values are stored. The bounding box width is calculated by finding the difference between right and left value.

Using these dimensions, boundary of the image is drawn. First value is the initial position of the pixel. Last value means the final position of the pixel.

$$\text{Height} = {(\text{bottom value} - \text{top value})}\big/{2} \qquad (4)$$

$$\text{Width} = {(\text{right value} - \text{left value})}\big/{2} \qquad (5)$$

The height value is added to the top value. It is then stored as middle_top. The width value is added to the left value and is stored as middle_left. The pixel at the pixel value at (middle_top, middle_left) is assigned as the maximum intensity to the pixel.

4.1.4 Distance between centroid and Velocity

The centroid of the object is used to find the distance travelled by the moving objects by using the Euclidean distance formula. The moving object's pixel positions at the first stage to the last stage are used as variables.

The centroid position of the object in each image is calculated. After this the distance between consecutive centroid images are obtained. Then by defining the first and the last value position of the X and Y resolution, change in distance is found using the formula:

$$\text{Distance} = \sqrt{(\mathrm{X2} - \mathrm{X1})^2 + (\mathrm{Y2} - \mathrm{Y1})^2} \qquad (6)$$

Here X1 is the earlier pixel position and X2 represents the current pixel position in width. Y1 is the earlier pixel position and Y2 represents the current pixel position in height.

For calculating velocity, the distance travelled by the moving object is read. Then the velocity in pixels / second is calculated. The value obtained is stored in an array.

4.2 Online mode

The steps in implementing the online mode algorithm are as given below:

1. Select a threshold value (sensitivity).

2. Trigger to start video acquisition.

3. Adjust the background noise using median filter and Gaussian smoothing operator.

4. The difference between every two successive frames is evaluated.

5. Check if the difference is greater than the threshold value.

6. If the condition is yes, motion detected and the snapshot of that movement is captured and the alarm sound is triggered.

7. If the condition is no, the checking process continues.

8. Even after the movement is detected, the checking process continues and for each and every new movement identified, corresponding snapshot is generated and alarm rings.

Finally, we have defined two metrics for characterizing the whole system - Detection Rate (DR) and the False Alarm Rate (FAR). These rates are used to quantify the output of the proposed system, based on:

TP (true positive): detected regions that correspond to moving objects.

FP (false positive): detected regions that do not correspond to a moving object.

FN (false negative): moving objects that are not detected.

These scalars are combined to define the following metrics and are reported in Table 2:

$$DR = \frac{TP}{TP+FN} \quad (7)$$

$$FAR = \frac{FP}{TP+FP} \quad (8)$$

As the number of moving objects is small and its size varies, these measurements may have large variances. This table shows that the moving objects detected at different instances, leading to a perfect detection rate in all examples.

5. SIMULATION RESULTS

Simulation of the proposed algorithm in offline and online mode is done in MATLAB. The calibration consists of the determination of the mapping between pixel plane of the camera and the 3D world directions. The intrinsic calibration maps each pixel to a unit vector in the camera reference and assumed known by any of the standard procedure. The extrinsic calibration gives the position and orientation of the camera reference with respect to a suitably chosen world reference frame as in Fig. 5.9.

5.1 Offline mode

The tracking of moving object is identified. The camera input frames of video at four various positions are shown in Fig.6.The pre-processed images, colour enhanced images, grey scale images, binary images of the video at the same four different positions are shown in the Fig. 7, 8, 9, 10 respectively. The final output with the label over the object is shown in the Fig.11. The labelled object is marked with a red colour rectangular box.

As the object is detected in the output video stream, the imposter detection is made known with the help of an alarm.

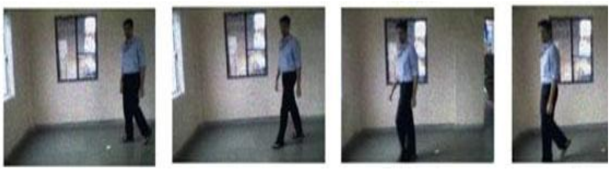


Fig 6: Different positions of object in input video

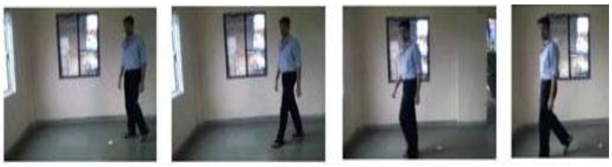


Fig 7: Noise removed images of frames

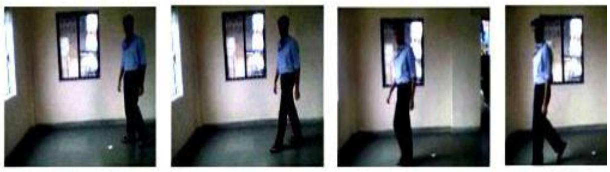


Fig 8: Colour enhanced images

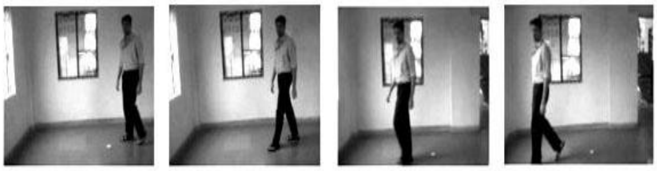


Fig 9: Grey scale conversion of images

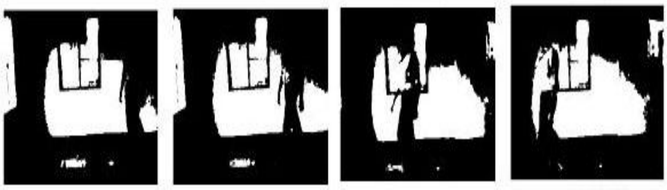


Fig 10: Binary images

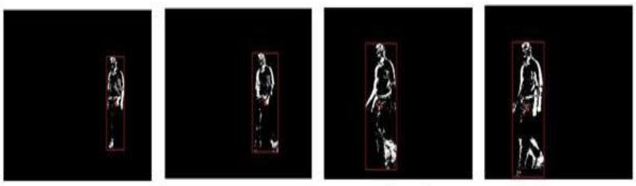


Fig 11: Detected object in output video with rectangular box

Here only one moving object is present and the distance between the centroid of moving object in each frame is calculated and thus the velocity of the moving object is found.

5.2 Online mode

The real time video capturing is done only after activating the mode using a trigger (here a activate button is used). The detection of movement of objects in two scenarios has been tested as shown in the Fig.12 and Fig.14.

Fig 12: Scenario for vehicle detection in unauthorised area

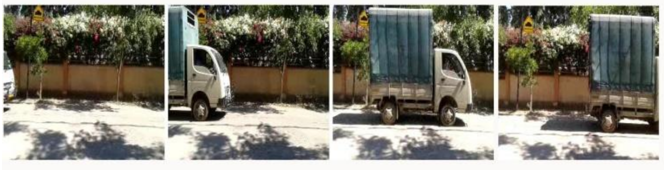


Fig 13: Snapshots generated as a result of vehicle detection

In the first scenario, Fig.12 and Fig.13 shows the snapshots generated during the testing of the algorithm for motion detection of the vehicle in an unauthorized area. And in the latter one, Fig.14 and Fig. 15 shows the snapshots generated during the testing of motion detection algorithm for intruder detection, to find the presence of a human which can be implemented in highly confidential and secured area.

In both the cases as soon as the initial movement is detected, the motion detection algorithm generates the snapshot of the movement and an alarm is activated. Thus it is verified that this algorithm is able to actively prevent/block intrusions and can improve security reinforcement in real time.

Fig 14: Scenario for intruder detection in highly confidential area

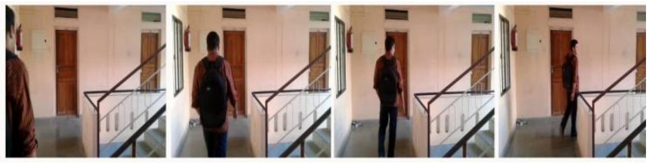


Fig 15: Snapshots generated as a result of intruder detection

The online algorithm is tested in other scenarios and the accuracy of object detection results are as shown in Table 1. For each scenario, the object detection test is done ten times and according to the result accuracy is determined. In the experiment conducted, even small sized moving objects were detected with higher accuracy rate. The detected object is highlighted in a red rectangle.

$$\text{Accuracy} = \frac{\text{Number of time the object detected}}{\text{Total number of time the test conducted}} \quad (9)$$

TABLE 1. Object detection accuracy test for different scenarios

Scenario	Accuracy	Number of objexts detected
	90%	2
	90%	1
	100%	1

6. CONCLUSION

The tracking of moving object in offline and online/real time mode is evaluated on MATLAB. The implementation of offline algorithm is done by removing unwanted noise, segmentation, feature extraction and labelling. The separation of foreground and background image is obtained by using the frame difference method. The object is tracked and the distance between the centroid of the object in each frame is stored to find the distance the object travelled and thereby its velocity of motion. Using of threshold value in the image segmentation plays an important role. The algorithm for online mode is useful in real time applications. The implementation is carried out by reducing the noise in the background, checking frame difference and generating the snapshot for each movement identified. For each movement detected alarm sound is triggered. Thus for the detection of moving objects in video a new algorithm is designed and implemented, which is useful in several applications like security, surveillance and vision analysis.

As future work the object area, size, shape and orientation in the image can be calculated from pixel values of moving object and this algorithm can be implemented on hardware devices. Also with the help of sensors like ultrasonic sensors on robot the distance between the object and the robot can be calculated and thus such modifications make it useful in complex robotic applications.

7. REFERENCES

[1] Gottipati. Srinivas Babu, “Moving Object Detection Using Matlab”, International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 6, August - 2012 .

[2] Vibha L, Chetana Hegde, P Deepa Shenoy, Venugopal K R, L M Patnaik, “Dynamic Object Detection, Tracking and Counting in Video Streams for Multimedia Mining” IAENG International journal of computer science ,August 2008.

[3] Kalpesh R Jadav, Prof.M.A.Lokhandwala, Prof.A.P.Gharge “Vision based moving object detection and tracking”, National Conference on Recent Trends in Engineering & Technology, 13-14 May 2011.

[4] Prof. Vinay U. Kale Mr. Pravin R. Lakhe “Moving Human Object Tracking, Estimation, Background Subtraction And Shadow Removal Technique” ,International journal of innovative research and studies, Volume 2 Issue 4,April 2013.

[5] Sandeep Bhatia , Mr. Ajay Mudgil ,Ms. Amita Son “Alive Human Body Detection system using a Autonomous Mobile Rescue Robot” International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 3, February 2012.

[6] Binoy Shah and Howie Choset,“Survey on Urban Search and Rescue Robotics,” Journal of Robotics Society of Japan, Vol 22, No. 5, Page 582-586, 2004.

[7] Vesna Zeljković, Željen Trpovski, Vojin Šenk, “New Algorithm for Moving Object Detection”, Yugoslav Journal of Operations Research 14, Number 1, 117-132, 2004.

[8] E. Grimson, C. Stauffer, R. Romano, and L.Lee, ªUsing Adaptive Tracking to Classify and Monitoring Activities in a Site, Proc. Computer Vision and Pattern Recognition Conf., pp. 22-29, 1998.

[9] I. Haritaoglu, D. Harwood, and L. Davis, W4:Who, When, Where, What: A Real Time System for Detecting and Tracking People, Proc. Third Face and Gesture Recognition Conf., pp. 222-227, 1998.

[10] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In Proceedings of the European Conference on Computer Vision, 1992.

[11] Kedar.A.Patwardhan, Guillerimo Saprio "Robust Foreground Detection in Video using pixel layers",IEEE Trans. Pattern Analysis and Machine Intelligence, Vol 30, no.4, April 2008.

[12] Chen S. C., Shyu M. L. and Zhang C., An Unsupervised Segmentation Framework for Texture Image Queries, In 25th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC), Chicago, Illinois, USA, Oct.2000.

[13] Chen S. C., Shyu M. L. and Zhang C., An Intelligent Framework for Spatio-Temporal Vehicle Tracking, 4th International IEEE Conference on Intelligent Transportation Systems, Oakland, California, USA, Aug. 2001.

[14] D. Beymer, P. McLauchlan, B. Coifman and J. Malik, A Real-time Computer Vision System for Measuring Traffic Parameters, In Proceedings of IEEE Conference Computer Vision and Pattern Recognition, Puerto Rico, pp. 496501,June, 1997.

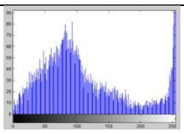
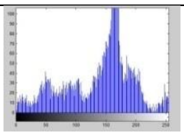
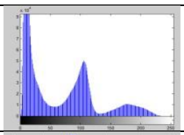
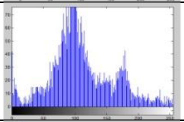
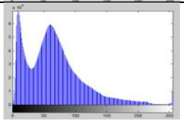
[15] N. P. Papanikolopoulos and O. Masoud, Robust Pedestrian Tracking Using a Model-based Approach, In Proceedings of IEEE Conference on Intelligent Transportation Systems, pp. 338343, Nov. 1997.

[16] A. J. Lipton, H. Fujiyoshi and R. S. Patil, Moving Target Classification and Tracking from Real-time Video, In Proceedings of IEEE Workshop Applications of Computer Vision, pp. 814, 1998.

[17] Gupte S., Masoud O., Martin R. F. K. and Papanikolopoulos N. P., Detection and Classification of Vehicles, In IEEE Transactions on Intelligent Transportation Systems, vol. 3, no.1, pp. 37 47,March 2002.

[18] Dailey D. J., Cathey F. and Pumrin S., An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras, In IEEE Transactions on Intelligent Transportations Systems, vol. 1, No. 2, pp.98-107, June, 2000.

TABLE 2. Analysis of the motion detection algorithms

Video Stream	No of Objects moving	Online /Offline	Time	Detection	Mean	Standard Deviation	Tracking		Performance	
			Day / Night	Detected Region			Region	Path	DR	FAR
	1	Online	Day		112.5976	69.2403	1	1	1.0	0.0
	1	Online	Day		138.7529	55.8919	1	2	1.0	0.0
	1	Online	Night		63.7034	59.3310	1	1	1.0	0.0
	1	Offline	Day		111.1007	49.7275	1	1	1.0	0.0
	3	Offline	Night		66.7763	45.1395	1	3	1.0	0.2