MOTION DETECTION AND VIDEO SURVILLANCE OF HUMAN BEING USING IMAGE PROCESSING

[1] Rashmi Kiran Nemade, [2] Dr.P.M.Mahajan

[1]P. G. Student, [2] Associate Professor, Department of Electronics and Telecommunication, Dean Academic J.T.Mahajan College of Engg, Faizpur, Maharashtra. India

[1] Department of Electronics and Telecommunication, J.T.Mahajan College of Engg, Faizpur. Maharashtra. India.

Abstract : This paper proposes a novel approach to create an automated visual surveillance system which is very efficient in detecting and tracking moving objects in a without any apriori information about the captured scene. The paper evaluates two methods for detection and tracking of moving objects. The background subtraction method is the common method of motion detection. It is a technology that uses the difference of the current image and the background image to detect the motion region, and it is generally able to provide data included object information. The background image is subtracted from the current frame. The second method created a motion-based system for detecting and tracking multiple moving objects. The detection of moving objects uses a background subtraction algorithm based on Gaussian mixture models. Morphological operations are applied to the resulting foreground mask to eliminate noise. Finally, blob analysis detects groups of connected pixels, which are likely to correspond to moving objects. The association of detections to the same object is based solely on motion. The motion of each track is estimated by a Kalman filter. The filter is used to predict the track's location in each frame, and determine the likelihood of each detection being assigned to each track. The performance is evaluated in terms of execution time for different videos under different circumstances. The performance evaluation results demonstrated that motion based video object detection method need more time as compared to background subtraction method. The background substraction method is more effective for detection and tracking of moving objects.

***IndexTerms* - visual surveillance system, background subtraction, motion-based system, Kalman filter**.

I. INTRODUCTION

Now-a-days, due to various security reasons surveillance systems became more popular and necessary. This enormous requirement increased the growth and technology improvement in tracking the moving the objects. It is widely used by the military, intelligence monitoring, human machine interface, virtual reality, motion analysis in tracking and detecting the object accurately. This increased requirement makes the research people to pay more intention developing an advanced methodology. Generally detection and tracking system based on two methods, in which the first one uses radar technology for tracking and another one is image processing technology by which it will capture the target tracking [1]. In this paper, we are indented to achieve better in result of detection and tracking. According to the image processing method various movements of the object is tracked from each frame along with its position information to get the various objectives Trajectory. In order to build a best video surveillance system the basic requirement is preparing an algorithm, with the motto of obtaining a result on the basis of fast, reliable and robust moving object detection and tracking system [2].

In this work we have evaluate the performance of two methods for detection and tracking of moving objects in video. The first method deals with detection and tracking of moving objects in video using background subtraction. In the second method we have used the Computer Vision Toolbox of MATLAB for performing motion based multiple moving object detection and tracking [3].

II. Literature Review

S.Kshirsagar et.al [4] proposes a system for moving object detection from video sequences. This work uses background subtraction method for real time application under conditions where illumination cannot be controlled. Tracking moving object in the cluttered scene is also done by background modelling i.e. technique to extract static initial background modelling model and later when the situation changes, knowledge segmentation and edge detection algorithm is used for modelling. The process of background updation at every frame is introduced in this work. This algorithm were applied on three video sequences. This algorithm which is invariant to external luminance changes, has been tasted under various lightning conditions, artificially simulated on the computer and with the moving object brighter and darker than the background. The performance parameter like MSE, entropy, precision and PSNR are calculated.

Benoit et.al [5] proposes Human visual system modelling for bio-inspired low level processing. This work is carried out on real time image processing module while they uses two parts of visual system. First one is retina and second is V1 cortex. The retina works as preprocessing step for high-level analysis while V1 cortex works as a low-level visual information. Retina level processing approach is used for head motion analysis with an application of hypo vigilance analysis to detect and prevent a driver from falling a sleep. This retina model is used for application of video surveillance because it is has ability to enhance visual information even in the case of back-lit situation and noise. This work also focuses on gray level image processing.

R.Estok et.al [6] proposes a system of human detection in image based on Intel Galilo. In this work comparative analysis and the methods for motion detection and human recognition is the image is made. Hardware and software solution is implemented for motion detection in the video with human recognition in the picture. This work compare two frames in bitmap image file which support RGB colour model. The proposed system consist of development board Intel Galileo and video source from IP camera are used as a basis for hardware implementation. This implemented solution is tested on two different scene. Each sequence for human recognition is tested under different conditions of sensitivity such as low, medium and high sensitivity to change colour value of a pixel: scene 1 for indoor and human recognition test results: scene 2 for storeroom. Out of 1200 images of scene 1 total 186 images were recognised and similarly out of 850 image of scene 2, total 141 images were recognised under various conditions and sensitivity.

K.Harish et.al [7] proposes smart video surveillance. They work on detecting multiple activities in real-time video. It is use for automatic video analytic to enhance effectiveness of surveillance system. This system also work on single person activity to enhance the security system in home video surveillance camera are used for security purpose.

III. BACKGROUND SUBSTRACTION METHOD FOR MOVING OBJECT DETECTION AND TRACKING

In proposed system the main aim is to build robust moving object detection algorithm that can detect and Track object in video. The background subtraction method is the common method of motion detection. It is a technology that uses the difference of the current image and the background image to detect the motion region, and it is generally able to provide data included object information. The background image is subtracted from the current frame. If the pixel difference is greater than the set threshold value T, then it determines that the pixels from the moving object, otherwise, as the background pixels. By using dynamic threshold method we can dynamically change the threshold value according to the lighting changes of the two images obtained [8]. This method can effectively suppress the impact of light changes. Here we consider first frame as the background frame directly and then that frame is subtracted from current frame to detect moving object. Background subtraction is the most widely used method for moving object detection. In this background subtraction method every pixel of on-going frame is subtracted with the pixels of the background image. After the background image B(x, y) is obtained, subtract the background image B(x, y) from the current frame Fk (x, y). If the pixel difference is greater than the set threshold value T, then determines that the pixels occur in the moving object, otherwise, as the background pixels. The moving object can be detected after applying threshold operation. As in the algorithm T is a fixed value, only for an ideal condition, is not suitable for complex environment with lighting changes.

1. The first step is to take input video from static cameras. For processing the video files, convert video into frames and from frames to images.
2. Next step is take first frame as a Background frame and next is current frame and then apply subtraction operation. Background frame is subtracted from current frame.
3. Then Threshold operation is performed and foreground object is detected.
4. After object detected last step is track object in video.

Figure 1 shows flow chart for moving object detection using reference Background. Reference Background means Background is fixed.

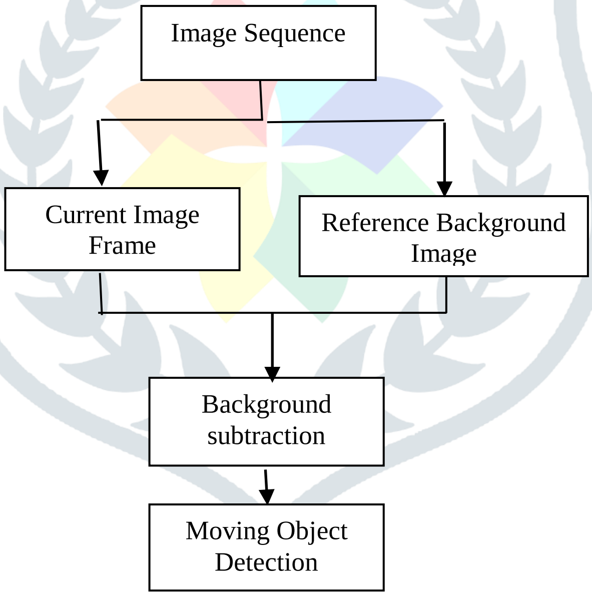

Figure 1: Background Subtraction for Moving Object Detection and Tracking

IV. MOTION BASED MULTIPLE OBJECT TRACKING

Detection of moving objects and motion-based tracking are important components of many computer vision applications, including activity recognition, traffic monitoring, and automotive safety. The problem of motion-based object tracking can be divided into two parts:

- detecting moving objects in each frame
- associating the detections corresponding to the same object over time

The detection of moving objects uses a background subtraction algorithm based on Gaussian mixture models. Morphological operations are applied to the resulting foreground mask to eliminate noise. Finally, blob analysis detects groups of connected pixels, which are likely to correspond to moving objects.

The association of detections to the same object is based solely on motion. The motion of each track is estimated by a Kalman filter. The filter is used to predict the track's location in each frame, and determine the likelihood of each detection being assigned to each track. In a motion-based system for detecting and tracking multiple moving objects [9]. The detection of moving objects uses a background subtraction algorithm based on Gaussian mixture models. Morphological operations are applied to the resulting foreground mask to eliminate noise. Finally, blob analysis detects groups of connected pixels, which are likely to correspond to moving objects. The kalman filter is used to predict the track's location in each frame, and determine the likelihood of each detection being assigned to each track.

Track maintenance becomes an important aspect of this method. In any given frame, some detections may be assigned to tracks, while other detections and tracks may remain unassigned. The assigned tracks are updated using the corresponding detections. The unassigned tracks are marked invisible. An unassigned detection begins a new track.

Each track keeps count of the number of consecutive frames, where it remained unassigned. If the count exceeds a specified threshold, the example assumes that the object left the field of view and it deletes the track [10].

The algorithm for motion based moving object detection consist of following steps

1. Create System Objects
   Create System objects used for reading the video frames, detecting foreground objects, and displaying results.
2. Initialize Tracks
   The initializeTracks function creates an array of tracks, where each track is a structure representing a moving object in the video. The purpose of the structure is to maintain the state of a tracked object. The state consists of information used for detection to track assignment, track termination, and display
3. Read a Video Frame
   Read the next video frame from the video file.
4. Detect Objects
   The detectObjects function returns the centroids and the bounding boxes of the detected objects. It also returns the binary mask, which has the same size as the input frame. Pixels with a value of 1 correspond to the foreground, and pixels with a value of 0 correspond to the background.
   The function performs motion segmentation using the foreground detector. It then performs morphological operations on the resulting binary mask to remove noisy pixels and to fill the holes in the remaining blobs.
5. Predict New Locations of Existing Tracks
   Use the Kalman filter to predict the centroid of each track in the current frame, and update its bounding box accordingly.
6. Assign Detections to Tracks
   Assigning object detections in the current frame to existing tracks is done by minimizing cost. The cost is defined as the negative log-likelihood of a detection corresponding to a track.
   The algorithm involves two steps:
   Step 1: Compute the cost of assigning every detection to each track using the distance method of the vision.KalmanFilter System object. The cost takes into account the Euclidean distance between the predicted centroid of the track and the centroid of the detection. It also includes the confidence of the prediction, which is maintained by the Kalman filter. The results are stored in an MxN matrix, where M is the number of tracks, and N is the number of detections.
   Step 2: Solve the assignment problem represented by the cost matrix using the assignDetectionsToTracks function. The function takes the cost matrix and the cost of not assigning any detections to a track.
   The value for the cost of not assigning a detection to a track depends on the range of values returned by the distance method of the vision.KalmanFilter. This value must be tuned experimentally. Setting it too low increases the likelihood of creating a new track, and may result in track fragmentation. Setting it too high may result in a single track corresponding to a series of separate moving objects.
   The assignDetectionsToTracks function uses the Munkres' version of the Hungarian algorithm to compute an assignment which minimizes the total cost. It returns an M x 2 matrix containing the corresponding indices of assigned tracks and detections in its two columns. It also returns the indices of tracks and detections that remained unassigned.
7. Update Assigned Tracks
   The updateAssignedTracks function updates each assigned track with the corresponding detection. It calls the correct method of vision.KalmanFilter to correct the location estimate. Next, it stores the new bounding box, and increases the age of the track and the total visible count by 1. Finally, the function sets the invisible count to 0.
8. Update Unassigned Tracks
   Mark each unassigned track as invisible, and increase its age by 1.
9. Delete Lost Tracks
   The deleteLostTracks function deletes tracks that have been invisible for too many consecutive frames. It also deletes recently created tracks that have been invisible for too many frames overall.
10. Create New Tracks
    Create new tracks from unassigned detections. Assume that any unassigned detection is a start of a new track. In practice, you can use other cues to eliminate noisy detections, such as size, location, or appearance.
11. Display Tracking Results
    The displayTrackingResults function draws a bounding box and label ID for each track on the video frame and the foreground mask. It then displays the frame and the mask in their respective video players.

V. EXPERIMENTAL ANALYSIS

We used Matlab software to develop the GUI shown in Fig.2.The GUI was designed to facilitate interactive system operation. GUI can be used to setup the program, launch it, stop it and display results. During select video stage the operator is promoted to choose the input video for motion detection and tracking. Whenever the start/stop toggle button is pressed the system will be launched and the selected program will be called to perform the calculations until the start/stop button is pressed again which will terminate the calculation and return control to GUI. Results can be viewed as detection and tracking consequently.

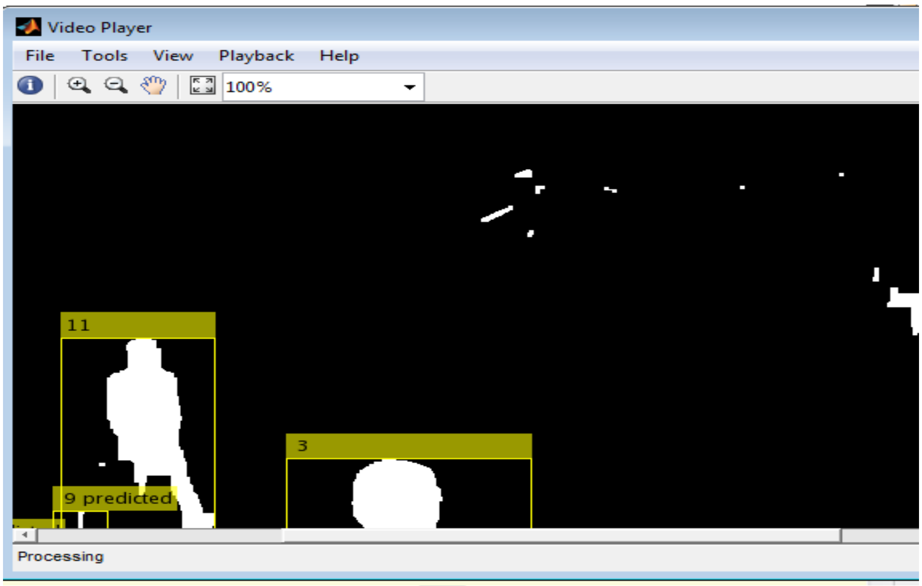

Figure 2: GUI Layout Design for background subtraction method

The motion based tracking algorithm is able to detect and track multiple moving objects.

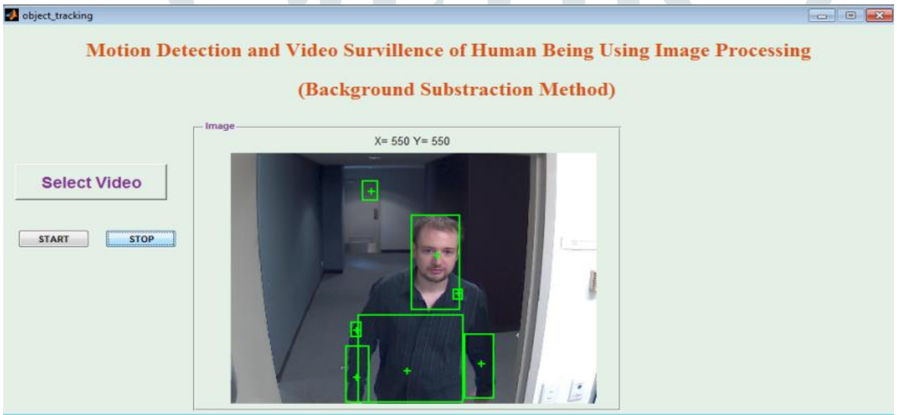

Figure 3: Motion based multiple objects detection and tracking

The method is able to detect and track multiple moving persons in the video along with the count showing number of persons detected and tracked.

For the selected frame, figure 4 shows the moving objects segmented and tracked.

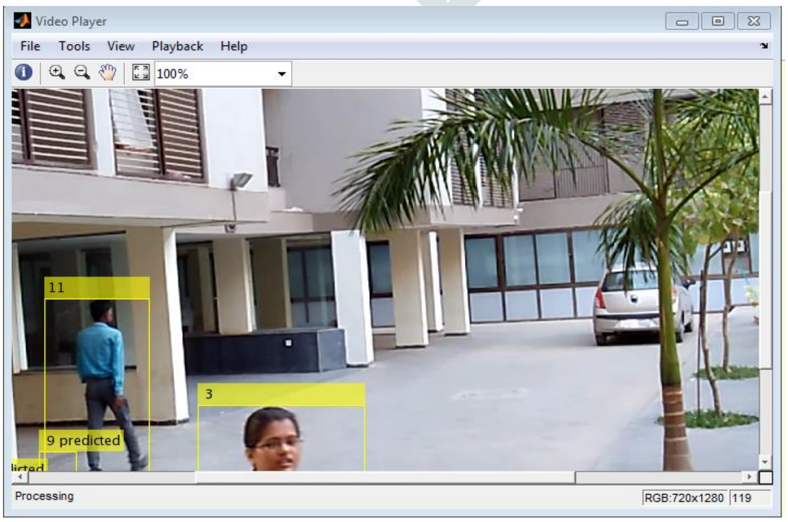

Figure 4: Moving Objects Segmentation and Tracking

We have evaluated the performance of both methods on 30 different videos collected from datasets. The dataset considered is ChokePoint Dataset [13]. The dataset is designed for experiments in person identification/verification under real-world surveillance conditions using existing technologies. An array of three cameras was placed above several portals (natural choke points in terms of pedestrian traffic) to capture subjects walking through each portal in a natural way. While a person is walking through a portal, a sequence of face images (ie. a face set) can be captured. Faces in such sets will have variations in terms of illumination conditions, pose, sharpness, as well as misalignment due to automatic face localisation/detection. Due to the three camera configuration, one of the cameras is likely to capture a face set where a subset of the faces is near-frontal.

The dataset consists of 25 subjects (19 male and 6 female) in portal 1 and 29 subjects (23 male and 6 female) in portal 2. The dataset has frame rate of 30 fps and the image resolution is 800X600 pixels. In total, the dataset consists of 48 video sequences and 64,204 face images. In all sequences, only one subject is presented in the image at a time. The first 100 frames of each sequence are for background modelling where no foreground objects were presented.

Table 1 shows the execution time for both methods. As depicted in the table, the motion based object tracking method requires more time as compared to background subtraction method. The object detection and tracking results are also better for the background subtraction method as compared to motion based moving object detection method.

Table 1 shows the execution time for both methods

Sr No.	Name of the Video	Execution Time in Seconds	
		Background Subtraction	Motion Based Moving Object Detection
1	Video 1	159.6215	69.1572
2	Video 2	210.1634	88.0174
3	Video 3	104.7065	36.3174
4	Car-overhead-1-hires	115.5637	118.4255
5	Car-overhead-2-hires	28.6745	33.0884
6	Car-overhead-3-hires	74.8992	77.7127
7	Car-perspective-1-hires	199.6021	207.7161
8	Car-perspective-2-hires	23.4208	54.9389
9	Car-perspective-3-hires	41.6576	35.2119
10	P1E-S1-C1	17.3403	27.0321
11	P1E-S2-C1	21.9698	23.5211
12	P1E-S3-C1	18.9058	28.6544
13	P1E-S4-C1	46.9285	48.2937
14	P1L-S1-C1	21.8259	48.8254
15	P1L-S2-C1	28.7843	23.9636
16	P1L-S3-C1	19.8529	21.4258
17	P1L-S4-C1	14.3085	50.2269
18	P2E-S1-C1.1	99.4301	116.1601
19	P2E-S2-C1.1	34.4267	50.0231
20	P2E-S3-C1.1	93.4390	138.3483
21	P2E-S4-C1.1	49.9537	57.5792
22	P2E-S5-C1.1	51.2324	65.9218
23	P2L-S1-C1.1	266.6929	430.6739
24	P2L-S1-C1.2	84.7658	85.6423
25	P2L-S1-C2.1	248.0585	85.6423
26	P2L-S2-C1.1	83.1290	368.6847
27	P2L-S3-C1.1	239.2811	372.3032
28	P2L-S4-C1.1	324.7754	374.3745
29	P2L-S5-C1.1	26.8889	26.9050
30	Tud-campus-sequence	11.2091	9.1843
	Average	**92.05**	**105.8**

II DISCUSSIONS AND CONCLUSION

Generally, we work on to develop an algorithm for moving object detection and tracking system. This algorithm is successfully implemented using Matlab integrated development environment. As a result, the algorithm is able to detect and track a moving object that is moving. In proposed system we presented a set of methods and tools for moving object detection and tracking system. We implemented two different object detection algorithms and evaluated the performance in terms of execution time. The Motion based moving object detection method need more execution time as compared to background subtraction method.

REFERENCES

1. D. Serby, E. K. Meier, and L. V. Gool "Probabilistic Object Tracking Using Multi ple Features " IEEE Proc. of International Conf on Pattern Recognition Intelligent Transportation Systems, Vol. 6, pp. 43- 53, 2004.
2. L. Li, S. Ranganath, H. Weimin, and K. Sengupta "Framework Framework for Real-Time Behavior Interpretation From Traffic Video " IEEE Tran. On Intelligen Transportation Systems, , Vol. 6, No. 1, pp. 43-53, 2005.
3. P. Kumar, H. Weimin, I. U. Gu, and Q. Tian "Statistical Modeling of Complex Backgrounds for Foreground Object Detection" IEEE Trans. On Image Processing, Vol. 13, No. 11, pp. 43-53, November 2004.
4. S.Kshirsagar, P.A.Ghonge,"Movement Detection Using Image Processing", International Journal Of Science and Research(IJSR), Vol.4 Issue 2,pp 2333,2015.
5. A.Benoit, A. Caplier, B.Durette, J.Herault,"Using Human Visual System Modelling fot bio-inspired low level image processing", Computer Vision and Image Understanding, pp 758-773.,2010.
6. R.Estok, O.Kainz, M.Michalko, F.Jakab,"System for Human Detection in Image Based on Intel Galileo", International Journal of Advanced Research in Artifical Intelligence, vol.4, No.9, pp.17-21, 2015.
7. K.Harish, G.Swati,K.Sonali,P.Sangmesh,"Smart Video Survillance", International Journal of Emerging Engineering Research and Technology, Vol.3, Issue 1, pp 109-112, 2015.
8. Z Zivkovi "Improving the selection of feature points for tracking" In Pattern Analysis and Applications, vol.7, no. 2, Copyright SpringerVerlag London Limited, 2004.
9. J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank "3D Model-Based Vehicle Tracking " IEEE Trans. on Image Processing, Vol. 14, pp. 1561-1569, October 2005.
10. Rachna Verma, "A Review of Object Detection and Tracking Methods", International Journal of Advance
11. Engineering and Research Development, vol 4, issue 10, pp569-578, Oct 2017.
12. C. S. Royden, K. D Moore, "Use Of Speed Cues In The Detection Of Moving Objects By Moving Observers," Vision research, Vol. 59, pp. 17-24, 2012.
13. SepehrAslani, Homayoun Mahdavi-Nasab, "Optical Flow Based Moving Object Detection and Tracking for Traffic Surveillance", World Academy of Science, Engineering and Technology International Journal of Electrical and Computer Engineering, vol. 7, No:9, 2013.
14. http://arma.sourceforge.net/chokepoint/#overview