Real Time Moving Object Tracking

Rajiv Ranjan
University at Buffalo
Buffalo, NY, USA
rajivran@buffalo.edu

Abstract

Monitoring continuously the movements of persons or vehicles and reporting when predefined events take place is a very common security application. A human observation based system for implementing this has several disadvantages. The present day technology allows automatic detection based on predefined measures. In this project a study on image processing toolkit using MATLAB has been conducted to perform moving object detection on video file. First, video pre-processing steps such as frame separation, binary operation and gray enhancement were conducted. Then the detection of moving object was carried out on video frames according to frame-difference based dynamic-background refreshing algorithm. Preliminary experimental results show that algorithm performs well. The results showed that the algorithm proposed achieve efficient detection and tracking of multiple moving objects in real time scenario. The algorithm proposed is validated on vehicle image sequence. With our approach, real-time level-set based video tracking can be achieved.

1. Introduction

In Imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Videos are actually sequences of images, each of which called a frame, displayed in fast enough frequency so that human eyes can percept the continuity of its content. It is obvious that all image processing techniques can be applied to individual frames. Besides, the contents of two consecutive frames are usually closely related.

This paper describes a novel way for tracking of a general purpose object which is fail-safe. Moving object tracking of video image sequences is one of the most important subjects in computer vision. It has already been applied in many computer vision fields, such as video surveillance, artificial intelligence, military guidance, safety detection, and robot navigation, medical and biological application etc.

1.1. Objectives

The objective of this project is to effectively detect a moving object and track it. For tracking the object Kalman Filter will be used. In this project, Kalman filter is used to establish object motion model, using the current object's information to predict object's position, so that we can reduce the search scope and search time of moving object to achieve fast tracking. For detecting the objects, we will be using various algorithms like background subtraction, edge detection etc. Finally tracking the object using many suitable algorithms and analyzing these algorithms.

2. Related Work/ Literature review

Here is presented a list of research papers referenced for this project. These papers have really helped in laying the foundation and have been the guiding path in the project.

A. **Multiple Object Tracking Method using Kalman Filter -Xin Li, Kejun Wang, Wei Wang and Yang Li (2010):** In this paper background subtraction method is used to detect and extract moving objects. Kalman filter model is used to handle multiple object tracking. It chooses centroid and size of tracking window as the feature value to describe moving object. It includes tracking results for a video of moving bodies in an indoor scene. The paper focuses on how prediction of Kalman filter is used for tracking an object.

B. Moving Object Detection using MATLAB- Gottipati Srinivas Babu (2012): In this paper background subtraction is used to detect object. The algorithm proposes to extract the background from all the frames of video and detects the foreground effectively. This algorithm updates the background frame dynamically and identifies the shadow of the moving object so as to calculate the area of the object accurately.

C. Object Tracking using Correlation, Kalman Filter and Fast Means Shift Algorithms- Ahmad Ali, Dr. Sikander Majid Mirza (2006): In this paper template-matching technique is used for object tracking. It uses normalized 2D cross correlation for template matching. In every next frame, template is matched. A threshold value is chosen for normalized cross correlation. To increase the efficiency of the algorithm, it works only in ROI instead of the whole scene. Kalman filter works parallel with template matching algorithm. As correlation value falls below threshold, then template is not updated but searching of object in the scene continues.

D. An Approach for Efficient Real Time Moving Object Detection -Arnab Roy, Sanket Shinde and Kyoung Don Kang (2008): This paper uses the edge detection technique using Sobel filter to detect objects. In this paper two algorithms are discussed. In 1st algorithm, first background subtraction technique is done and then Sobel operator is applied whereas in 2nd algorithm first Sobel operator is applied to detect edges and then background subtraction is done. In background subtraction, image background and foreground are separated and analyzed. The data found from it is then used to detect motion. We run Sobel filtering on the image in order to remove noise and to detect the edges. In addition, we perform mean or median filtering to take care of potential noise.

E. Application of MATLAB in moving object detection algorithm, International Seminar on Future Biomedical Information Engineering, 2008. Xiao Chen: It discusses the use Matlab for various object detection algorithms and analyses the usefulness of tools provided. Matlab has the characteristics of simple programming, easy operation and high processing rate, etc. when used in series of processing of moving object detecting algorithm. The image processing toolkit for language Matlab has powerful functions, with which all commonly used techniques and methods in image processing can be implemented.

F. "Implementation of Object Tracking for Real Time Video", ACEEE Int. J. on Communications, Vol. 4, No. 1, July 2013 Rajesh Chhikara, Sunil Kumar, Gopinath S: This paper takes a look at the level set method for handling topological changes such as the merging and splitting of object regions while performing real-time tracking of object boundaries. It has proven to be a more powerful technique and various models have been proposed.

3.Methodology/ Algorithm Implemented

Below is detailed explanation of the methodology used for this project.

3.1 Background Theory

A video sequences consists of frame sequences, the detection for moving object in video is conducted in a way that frame sequences are extracted from video sequence. Therefore, moving object detecting has something similar to object detection in still images. An image representation is two dimensional (2D), although it corresponds to one of the three-dimensional (3D) object or scene.

In this project, term image processing to refer to all the techniques and applications described in project, whether the output is a modified (i.e., processed) version of the input image, an encoded version of its main attributes, or a non pictorial description of its contents. Moreover, three levels of image processing operations were distinguished as:

- Low Level: Primitive operations (e.g., noise reduction, contrast enhancement, etc.) where both the input and the output are images.
- Mid Level: Extraction of attributes (e.g., edges, contours, regions, etc.) from images.
- High Level: Analysis and interpretation of the contents of a scene.

The method used in moving object detecting is video sequence analysis. Two or more frames acquired at different time contain the information about relative motion between an imaging system and scene. This information is in forms of the color variation between frames. Therefore, the information about motions can be obtained through analysis and processing of images acquired at different time. Video sequence analysis methods can be classified into three types: background difference method, adjacent frame difference method and edge detection method.

In adjacent frame difference method, moving object is extracted according to differences among two or three continuous frames. The method is the simplest and direct, within which the changing part in video can be quickly detected. Edge detection is a type of image segmentation techniques which determines the presence of an edge or line in an image and outlines them in an appropriate way. The main purpose of edge detection is to simplify the image data in order to minimize the amount of data to be processed.

A. Object detection in consequent images is nothing but the detection of the moving object in the scene. In video surveillance, object detection refers to the capability of the surveillance system to detect motion and capture the events. Object detection is usually a software-based Monitoring algorithm which will signal the surveillance camera to begin capturing the event when it detects motions. This is also called activity detection.

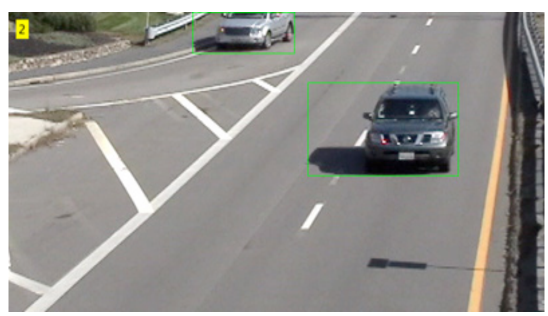

Figure 1: Moving cars are detected by blob analysis

B. **Background Subtraction:** One common technique to do foreground segmentation when having a stationary camera is background subtraction modelling. In this method, generating a reference background image and simply comparing a current image with this reference background image. Generate a reference background image by averaging all corresponding pixel intensities for every frames using 1.

$$BG(x,y) = \frac{\Sigma_{m=1}^{N} I(x,y)}{N}$$

The pixel values of the current frame are compared with the corresponding pixel values of the fixed reference background image using 2. The parts where the images do not match are considered as a foreground.

C. Edge Detection

Edge detection is the name for a set of mathematical methods which aim at identifying the points in a digital image at which image brightness changes sharply or more formally has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges.

Figure 2: Edge Detection on a human face

Kalman Filtering: Mathematically, Kalman filer is an estimator that predicts and corrects the states of wide range of linear processes. It is not only efficient practically but attractive theoretically as well precisely, the optimal state is found with smallest possible variance error, recursively. However, an accurate model is an essential requirement.

In Kalman filer, we consider a tracking system where x_k is the state vector which represents the dynamic behavior of the object, where subscript k indicates the discrete time. The objective is to estimate x_k from the measurement z_k.

1) Process Equation: xk =Axk-1 + wk-1

Where A represents the transition matrix and xk the state at time k -1 to k. Vector wk is the Gaussian process noise.

2) Measurement equation: Zk = Hxk + vk

Where H is the measurement matrix and Zk is the measurement observed at time k -1 to k respectively. vk is the Gaussian measurement noise.

3) Time update equations:

$$\hat{x}_k^- = A\hat{x}_{k-1} + w_k$$

$$P_k^- = AP_{k-1}A^T + Q$$

Equation (1) and (2) describes a linear model at time k. As xk is not measured directly, therefore the information provided by measured zk is used to update the unknown states xk estimate of state ˆx-k and covariance error ˆxk estimate is obtained for the next time step k.

4) Measurement update equations:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-)$$

$$P_k = (1 - K_k H) P_k^-$$

K is the Kalman gain which is computed by above the measurement update equations. After that state estimate ˆxk and error estimate Pk is computed by the measurement zk. The time and measurement equations are calculated recursively with previous estimates to predict new estimate. This recursive behavior of estimating the states is one of the highlights of the Kalman filter.

The algorithm works in 2 step process. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. Because of the algorithm's recursive nature, it can run in real time using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.

The Kalman filter model assumes the true state at time k is evolved from the state at (k−1) according to

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$$

where

Fk is the state transition model which is applied to the previous state xk−1;

Bk is the control-input model which is applied to the control vector uk;

wk is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance Qk.

In this project, work has been done on a traffic video in which moving cars have been tracked. Consider car infinitely long straight rails. Initially the truck is stationary at position 0, but it is buffeted this way and that by random acceleration. We measure the position of the truck every Δt seconds, but these measurements are imprecise; we want to maintain a model of where the truck is and what its velocity is. We show here how we derive the model from which we create our Kalman filter. Since F, H, R and Q are constant, their time indices are dropped.

3.2 Methodology

Initially a lot of research papers were considered for various object detection and object tracking techniques. Then the first part of project, which was object detection, was implemented.

3.2.1 Object Detection

First the technique of adjacent frame subtraction was implemented. Then two versions of edge detection technique using Sobel operator were implemented. And finally results of the above discussed techniques were compared for the best algorithm to be used in the next phase of project that was object tracking.

Finally, at last step of the second phase of the project, the adjacent frame subtraction technique was found to be the best suited to be applied into the object tracking.

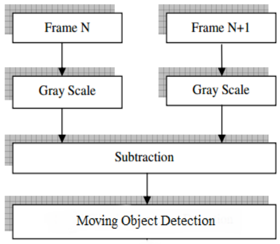

Figure 3: Adjacent Frame Subtraction Flowchart

Object detection using Sobel filter (Edge Detection based technique)

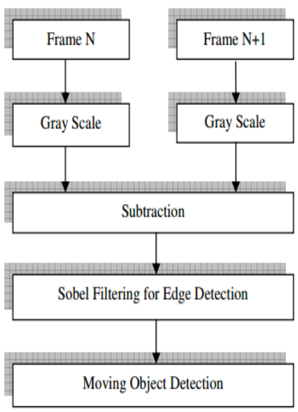

Figure 4: Edge Detection using Sobel operator

3.2.2 Object Tracking

Second phase of the project was object tracking. This was the most important phase of the object. In this part base paper was implemented and real time object were tracked using Kalman filter. Before implementing Kalman filter on real time videos, initially the simple video tracking was implemented. Then the concepts of video tracking were applied to real time video acquisition. After getting familiar with simple color based tracking techniques, focus was turned on to base paper that explains the implementation of Kalman filter. A simple Kalman filter was implemented on videos and moving objects were tracked. After achieving success in video tracking, Kalman filter was applied to real time videos.

At last a simple comparison was made between the simple color based tracking technique and the Kalman filtering technique.

3.3 Implementation Details

Below are the details of the implementation of the various algorithms.

3.3.1 Object Detection: We will first start with the object detection step.

3.3.1.1 Firstly, let's see the most basic image operations associated with an image.

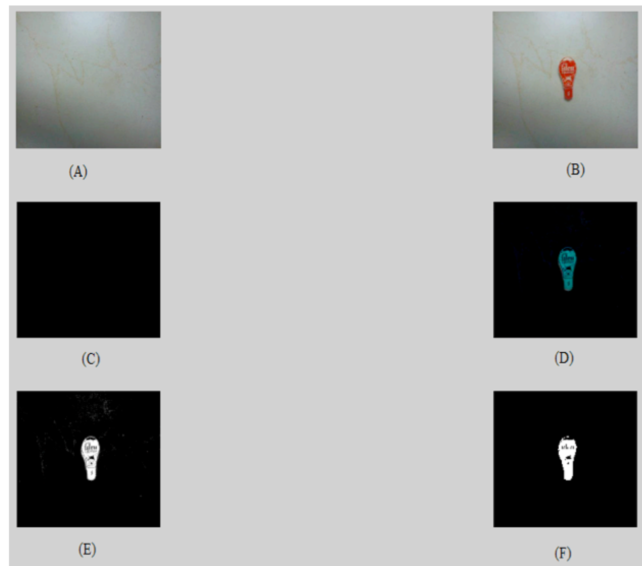

Figure 4: (A)Reference image (B) Image with object(C)In absence of object(D)Object being detected(E)Improving contrast by using inbuilt function (F) Binary form of (E).

This analysis uses a reference image as shown above. It illustrates how we enhanced an image, and then used it to identify the object. First read and display the reference image. Then create and display the gray scale image. To create a more uniform background, we subtracted the background image, from the original image, and then viewed the image. After subtraction, the image had a uniform background but was a bit too dark. Using imadjust, adjust the contrast of the image. Imadjust increases the contrast of the image by saturating 1% of the data at both low and high intensities of image and by stretching the intensity values to fill the uint8 dynamic range. Then create a binary version of the image so that toolbox functions can be on image. Using the im2bw function convert the gray scale image into a binary image by using thresholding. The function graythresh automatically computed an appropriate threshold value to convert the gray scale image to binary.

3.3.1.2 Object detection based on Sobel filter

First import a video file in the Matlab Workspace using the "VideoReader" function. Convert the video into frames, while reading the video through object created using function "VideoReader". Now the image sequence is in MATLAB, the simple technique for detecting the vehicles: frame differencing is used. To begin, convert each frame to gray scale using "rgb2gray". Running the loop "backwards," from numframes down to 1. The function graythresh computes a threshold that divides an image into background and foreground pixels. Since graythresh returns a normalized value in the range [0, 1], scale it to fit data range [0,255]. Subtract the two consecutive frames after converting it to gray scale using "imabsdiff". The absolute difference between successive frames can be used to divide an image frame into changed and unchanged regions. Since only the vehicles move, changed region was associated only with the vehicle, or possibly with its shadow. Next we used the Sobel operator to extract boundaries of object in each frame by running the loop "backwards," from numframes down to 1.

3.3.1.3 Adjacent frame differentiation technique

First import a video file in the Matlab Workspace using the "VideoReader" function. Converted the video into frames, while reading the video through object created using function "VideoReader". Now the image sequence is in MATLAB, the simple technique for detecting the vehicles: frame differencing is used. To begin, convert each frame to gray scale using "rgb2gray". Running the loop "backwards," from numframes down to 1. The function graythresh computes a threshold that divides an image into background and foreground pixels. Since graythresh returns a normalized value in the range [0, 1], scale it to fit data range [0,255]. Subtract the two consecutive frames after converting it to gray scale using "imabsdiff". The absolute difference between successive frames can be used to divide an image frame into changed and unchanged regions. Since only the vehicles move, changed region was associated only with the vehicle, or possibly with its shadow. The subtracted frames were converted to binary for the detection purpose of the moving vehicles using "bwareaopen". Again the frames were played in a loop downwards from numframes down to 1.

3.3.2 Object Tracking

3.3.2.1 Simple Color Based Video Tracking

Algorithm:

1. Create an object of stored video.
2. Read video into MATLAB.
3. Create a structure of frames.
4. Read all the frames into that structure.
5. Average first 10 frames and store it as the background image.
6. Initialize the values of centroids of each frame in video and a threshold value.
7. Compute the absolute mean of RGB color between current frame and background.
8. Convert the image obtained by subtraction of these frames into binary image.
9. Mark all the regions of interest using "bwlabel" and "regionprops".
10. Find the largest region using bubble sort.
11. Enclose this region into a bounding box.

3.3.2.2 Simple Real Time Tracking

Algorithm:

1. Create a video object for video input.
2. Set properties for the video object.
3. Start acquiring video frames.
4. Store first frame as background image.
5. While Frames acquired less than equal to 100

{Acquire current frame;
Subtract background from current frame;
Convert diff in both images to greyscale image;
Enclose the difference in both images by a bounding box;}

6. Stop acquiring video frames.

3.3.2.3 Object tracking using Kalman Filter (Single Object)

Algorithm:

1. Create an object of stored video.
2. Read video into MATLAB.
3. Create a structure of frames.
4. Read all the frames into that structure.
5. Average first 10 frames and store it as the background image.
6. Initialize the values of centroids of each frame in video and a threshold value.
7. For all video frames
8. Compute the absolute mean of RGB color between current frame and background.
9. Convert the image obtained by subtraction of these frames into binary image.
10. Mark all the regions of interest using "bwlabel" and "regionprops".
11. Find the largest region using bubble sort.
12. Find the size of Kalman window.
13. Initialize the predicted centroid and velocity for Kalman window size.
14. Use the former state to predict the new centroid and velocity.
15. Plot the tracking rectangle after Kalman filtering – red.
16. End for loop

3.3.2.4 Moving Object tracking using Kalman Filter (Multiple objects).

Algorithm:

1. Create an object of stored video.
2. Read video into MATLAB.
3. Create a structure of frames.
4. Read all the frames into that structure.
5. Average first 10 frames and store it as the background image.
6. Initialize the values of centroids of each frame in video and a threshold value.

7. For all video frames
8. Compute the absolute mean of RGB color between current frame and background.
9. Convert the image obtained by subtraction of these frames into binary image.
10. Mark all the regions of interest using “bwlabel” and “regionprops”.
11. If the area of the region of interest is greater than 100 pixels.
12. Find the size of Kalman window.
13. Initialize the predicted centroid and velocity for Kalman window size.
14. Use the former state to predict the new centroid and velocity.
15. Plot the tracking rectangle after Kalman filtering – red.
16. End if
17. End for loop

3.3.2.5 Object tracking using Kalman Filter (Single Object) Real Time using Webcam

Algorithm:

1. Create a video object for video input.
2. Set properties for the video object.
3. Start acquiring video frames.
4. Store first frame as background image.
5. While Frames acquired less than equal to 100

{Acquire current frame;
Subtract background from current frame;
Mark all the regions of interest using “bwlabel” and “regionprops”.
Do bubble sort (large to small) on regions in case there are more than 1 moving object detected.
The largest region is the object for consideration.
Plot the rectangle of background subtraction algorithm – blue.
Initialize the predicted centroid and velocity for Kalman window size.
Use the former state to predict the new centroid and velocity.
Plot the tracking rectangle after Kalman filtering – red}

6. Stop acquiring video frames.

3.3.2.6 Moving Object tracking using Kalman Filter (Multiple objects).

Algorithm:

1. Create a video object for video input.
2. Set properties for the video object.
3. Start acquiring video frames.
4. Store first frame as background image.
5. While Frames acquired less than equal to 100

{Acquire current frame;
Subtract background from current frame;
Mark all the regions of interest using “bwlabel” and “regionprops”.
If area of the region of interest is greater than 1000 pixel then
Plot the rectangle of background subtraction algorithm – blue.
Initialize the predicted centroid and velocity for Kalman window size.
Use the former state to predict the new centroid and velocity.
Plot the tracking rectangle after Kalman filtering – red
End if}

6. Stop acquiring video frames.

4. **Experimental analysis & Results**

4.1 Object Detection

Both the above mentioned algorithm for Object detection was implemented and executed and after analyzing the output, we saw that some stationary objects are also being detected along with moving objects. So we applied median filter to remove small stationary objects which resulted in distortion of moving objects also as can be seen above in frame no.73.

4.1.1 Object Detection by Adjacent Frame Subtraction

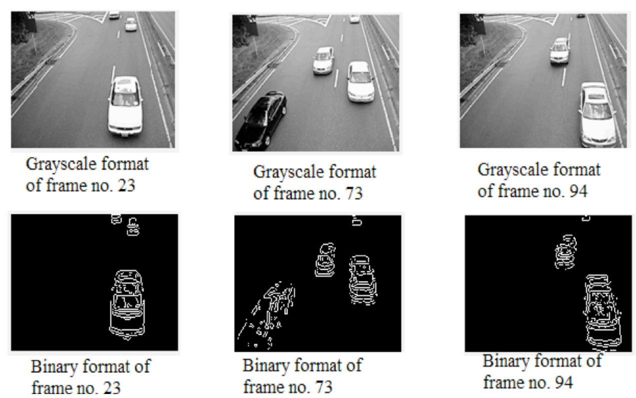

Figure 5: Snapshot of video taken at different time instant

4.1.2 Object Detection Based On Edge Detection using Sobel operator

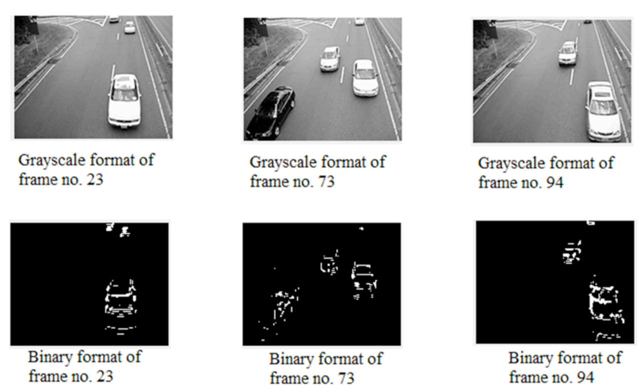

Figure 6: Snapshot of video taken at different time instant

Table 4.1.3 Analysis of Result Obtained of Detection

	ADJACENT FRAME DIFFERENTIATION ALGORITHM	OBJECT DETECTION USING DERIVATIVE OPERATOR-ALGORITHM 1	OBJECT DETECTION USING DERIVATIVE OPERATOR-ALGORITHM 2
EXECUTION TIME (Mean of 5 executions)	0.9073 seconds	1.5489 seconds	1.6043 seconds
QUALITY OF OBJECT DETECTED	Moving objects are detected clearly.	Moving objects are detected but are distorted to some extent.	Moving objects are detected but are greatly distorted.

Table Showing result of detection step for traffic video

4.2 Object Tracking

4.2.1 Object tracking in stored video using Kalman Filter (Single object)

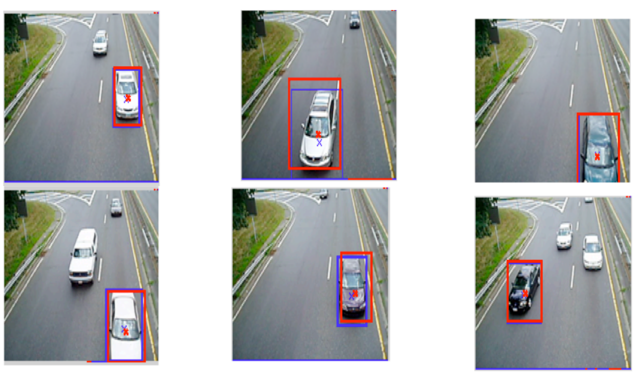

Figure 6: Tracking of vehicles using Kalman filter

The algorithm for Object tracking in stored video using Kalman Filter (Multiple objects) was executed and the results are shown above. The largest region is taken as the object for consideration. Plot the rectangle of background subtraction algorithm-blue. Initialize the predicted centroid and velocity for Kalman window size and use the former state to predict the new centroid and velocity. Plot the tracking rectangle after Kalman filtering-red.

4.2.2 Object tracking in stored video using Kalman Filter (Multiple objects)

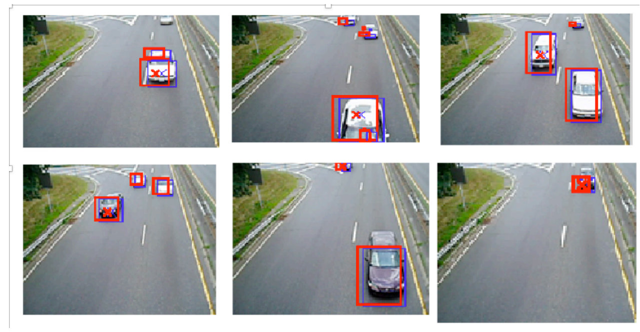

Figure 6:Tracking of multiple vehicles using Kalman filter

The algorithm for Object tracking in stored video using Kalman Filter (Multiple objects) was executed and after analyzing the output, it is observed that the vehicles tracked are better than median filter because it predicted the next movement of the vehicle from the previous frame that what its motion will be rather than just subtracting the two frames and obtaining the moving object and tracking the object. But the drawback is its execution time.

4.2.3 Object tracking in real time video using Kalman Filter using Webcam

The previously mentioned algorithms for Object tracking in real time video using Kalman Filter using Webcam is implemented for both single and multiple objects and executed. The results are shown below. The regions greater than 1000 pixels are only the Object for consideration. After analyzing the output, it is observed that this algorithm tracks multiple objects at the same time. Its tracking quality is better due to the prediction of the motion of the moving object, if sometimes the input is not clear or due to some technical fault.

4.2.3.1 For Single Object

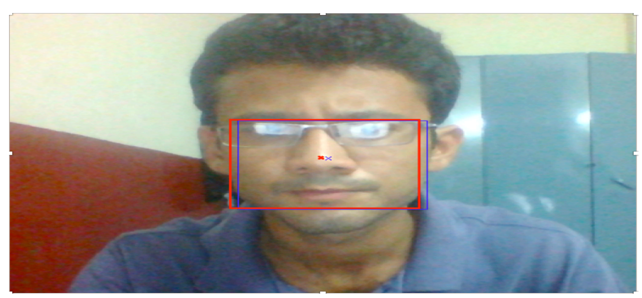

Figure 7: Tracking of real time single object using Kalman filter, of myself using a web cam at my home

4.2.3.1 For Multiple Objects

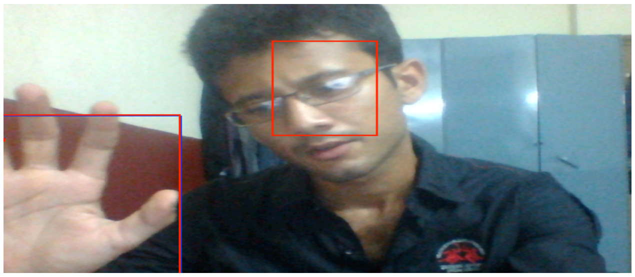

Figure 8: Tracking of real time multiple objects using Kalman filter, of myself using a web cam at my home

4.3 Tracking results for some other videos

4.3.1 The above mentioned various algorithms for object tracking were executed on a video called **car.mp4**. The results are below. The video was taken from YouTube.

1) Object tracking using Kalman Filter (Single object)

Figure 8: Snapshot of video taken at different time instant

2) Object tracking using Kalman Filter (Multiple objects)

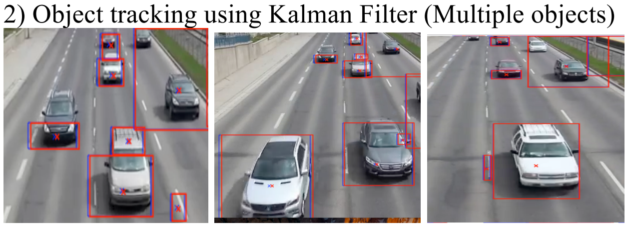

Figure 9: Snapshot of video taken at different time instant

4.3.2 Meet_Crowd.mp4. Source is CAVIAR test case.

1) Object tracking using Kalman Filter (Single object)

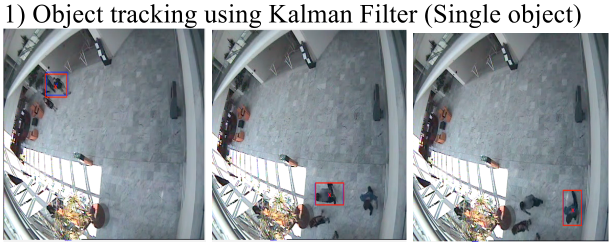

Figure10: Snapshot of video taken at different time instant

2) Object tracking using Kalman Filter (Multiple objects)

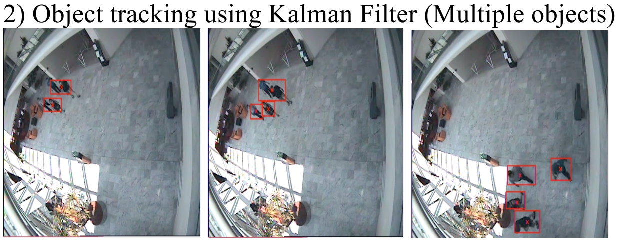

Figure11: Snapshot of video taken at different time instant

	ADJACENT FRAME DIFFERENTIATION ALGORITHM	OBJECT DETECTION USING DERIVATIVE OPERATOR-ALGORITHM 1	OBJECT DETECTION USING DERIVATIVE OPERATOR- ALGORITHM 2
1)EXECUTION TIME (for car.mp4) (Mean of 5 executions)	1.6903 seconds	2.1980 seconds	2.2674 seconds
2)EXECUTION TIME (for meet_crowd.mp4) (Mean of 5 executions)	1.9034 seconds	2.5056 seconds	2.6980 seconds
QUALITY OF OBJECT DETECTED	Moving objects are detected clearly.	Moving objects are detected but are distorted to some extent.	Moving objects are detected but are greatly distorted.

Table 4.3.3 showing detection step for 2 videos discussed

5 Discussion and conclusions:

5.1 Discussion

The main objective of this project is to efficiently detect and track real time moving objects using various algorithms with the help of MATLAB software. Methods like edge detection, background subtraction and adjacent frame subtraction detection techniques have been used to successfully detect the object. **Among these algorithms adjacent frame subtraction technique gave the best output.** The adjacent frame and background subtraction technique along with Kalman filter has been used to track the object.

5.2 Conclusion

Moving object detection is a basis for a number of important applications such as real time surveillance and visual tracking. However, it is computationally expensive and resource hungry. In this project three cost effective algorithms have been presented for real time moving objects detection. For performance evaluation the algorithms as well as a well-accepted approach for moving object detection that is based on background subtraction technique, have been implemented and evaluated.

5.2 Future Aspects and Further Improvements

In the future we aim to further improve object tracking algorithms. The performance and accuracy of the methods could be improved if we use other methods like Lucas-kanade and Mean-Shift Technique for object tracking.

References

[1] Xin Li, Kejun Wang, Wei Wang and Yang Li, "A Multiple Object Tracking Method Using Kalman Filter" Proceedings of the June, 2010

[2] Rajesh Chhikara, Sunil Kumar, Gopinath S, "Implementation of Object Tracking for Real Time Video"

[3] Gottipati. Srinivas Babu, "Moving object detection using MATLAB", International Journal of Engineering Research

[4] Xiao Chen, "Application of MATLAB in moving object detection algorithm", International Seminar on Future Biomedical Information Engineering, 2008.

[5] Arnab Roy, Sanket Shinde and Kyoung-Don Kang, An Approach for Efficient Real Time Moving Object Detection University of New York at Binghamton