



**RESEARCH ARTICLE**

# Moving Object Tracking Using Kalman Filter

**Hitesh A Patel<sup>1</sup>, Darshak G Thakore<sup>2</sup>**

<sup>1</sup>PG Scholar, Computer Engineering Department, BVM Engineering College, VV Nagar-388120, India

<sup>2</sup>Asso. Professor, Computer Engineering Department, BVM Engineering College, VV Nagar-388120, India

<sup>1</sup> [hiteshpatelbbit@yahoo.co.in](mailto:hiteshpatelbbit@yahoo.co.in); <sup>2</sup> [darshak\\_thakore@rediffmail.com](mailto:darshak_thakore@rediffmail.com)

---

**Abstract**— *In the field of security automated surveillance systems are very useful. Surveillance system can be used to detect and track the moving objects. First phase of the system is to detect the moving objects in the video. Second phase of the system will track the detected object. In this paper, detection of the moving object has been done using simple background subtraction and tracking of single moving object has been done using Kalman filter. The algorithm has been applied successfully on standard surveillance video datasets of CAVIAR [6] and PETS [7]. The videos used for testing have been taken using still cameras, which are located in indoor as well as outdoor environment having moderate to complex environments.*

**Key Terms:** - Visual surveillance; Kalman filter; Object detection; Background subtraction; Object tracking

---

## I. INTRODUCTION

Visual surveillance systems have been in use to monitor security sensitive areas. The availability of high-powered computers, high quality video cameras, and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms. There are three key steps in video analysis: detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of object tracks to recognize their behavior [5].

Video tracking can be defined as an action which can estimate the trajectory of an object in the image plane as it moves within a scene. A tracker assigns consistent labels to the tracked objects in different frames of a video. Tracking of an object can be done by continuously detecting to localize regions, points or features of an image frame by frame [2].

Second section of the paper covers description of the Kalman filter. Third section covers steps to implement Kalman filter for moving object tracking. Fourth section covers results of object tracking. Final section includes conclusion and future work.

## II. TRACKING ALGORITHM

Once the object has been detected then it can be tracked along its path. Many standard methods are available for object tracking. Viz. Kalman filter, Particle filter, Mean-shift based kernel tracking etc. Detail description of Kalman filter method is given below.

### 1) Kalman Filter

Kalman filter technique is used to estimate the state of a linear system where state is assumed to be distributed by a Gaussian [5]. In 1960, R.E. Kalman [1] published his famous paper describing a recursive solution to the discrete-data linear filtering problem [1]. Object tracking is performed by predicting the object's position from the previous information and verifying the existence of the object at the predicted position.

Secondly, the observed likelihood function and motion model must be learnt by some sample of image sequences before tracking is performed [2].

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process in several aspects: it supports estimations of past, present, and even future states, and it can do the same even when the precise nature of the modelled system is unknown [3].

The Kalman filter estimates a process by using a form of feedback control. The filter estimates the process state at some time and then obtains feedback in the form of noisy measurements. The equations for Kalman filters fall in two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimate for the next time step. The measurement update equations are responsible for the feedback. That is used for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. Figure 1 shows the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems. The time update projects the current state estimate ahead in time. The measurement update adjusts the projected estimate by an actual measurement at that time [3].

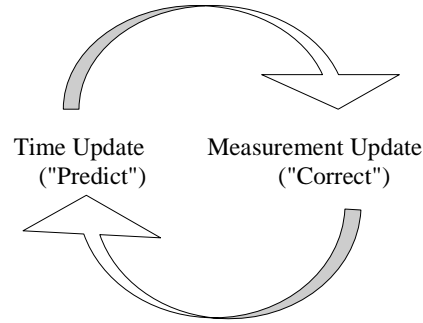


Fig 1. Discrete Kalman filter cycle [3]

The time update projects the current state estimate ahead in time. Time update equations[3,4]:

$$\text{State Prediction } X_{\text{pred}_k} = A * X_{k-1} + B * U_k + W_{k-1} \quad (1)$$

$$\text{Error Covariance Prediction } P_{\text{pred}_k} = A * P_{k-1} * A^T + Q \quad (2)$$

In eq. (1)  $X_{\text{pred}_k}$  is vector representing predicted process state at time  $k$ .  $X$  is a 4-dimensional vector  $[x \ y \ dx \ dy]$ , where  $x$  and  $y$  represent the coordinates of the object's centre, and  $dx$  and  $dy$  represent its velocity.

$X_{k-1}$  is vector representing process state at time  $k-1$ .  $A$  is a 4x4 process transition matrix of the form[4].

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$U_k$  is a control vector and  $B$  relates optional control vector  $U_k$  into state space.  $W_{k-1}$  is a process noise. In eq. (2)  $P_{\text{pred}_k}$  is predicted error covariance at time  $k$ .  $P_{k-1}$  is a matrix representing error covariance in the state prediction at time  $k-1$ , and  $Q$  is the process noise covariance[4].

Measurement update equations [3,4]:

After predicting the state  $X_{\text{pred}_k}$  and its error covariance at time  $k$  using the time update steps, the Kalman filter next uses measurement to correct its prediction during the measurement update steps[4].

$$\text{Kalman Gain } K_k = P_{\text{pred}_k} * H^T * (H * P_{\text{pred}_k} * H^T + R)^{-1} \quad (3)$$

$$\text{State Update } X_k = X_{\text{pred}_k} + K_k * (Z_k - H * X_{\text{pred}_k}) \quad (4)$$

$$\text{Error Covariance Update } P_k = (I - K_k * H) * P_{\text{pred}_k} \quad (5)$$

In eq. (3)  $K_k$  is Kalman gain.  $H$  is matrix converting state space into measurement space and  $R$  is measurement noise covariance. Determining  $R_k$  for set of measurement is difficult, many Kalman implementations statistically analyse training data to determine fixed  $R$  for all future time updates. In eq. (4)  $X_k$  is a process actual state. Using Kalman gain  $K_k$  and measurement  $Z_k$  process state  $X_k$  can be updated. Here  $Z_k$  is the most likely  $x$  and  $y$  coordinates of the target objects in the frame. The final step in Kalman filter is to update the error covariance  $P_{\text{pred}_k}$  into  $P_k$  as given in eq. (5). After each time and measurement update pair, the process is repeated with previous posteriori estimates used to project or predict the new priori estimate[4].

### III. IMPLEMENTATION

Tracking of moving object has been done using Kalman filter. Here tracking of any object can be done by providing the frame number from which tracking has to be started. From the selected frame any object can be picked for tracking by setting the position of the mask and then the object can be tracked in subsequent frames. Following steps have been implemented for tracking a single object.

- Background frame has been calculated by taking average of all the pixels.
- Frame number has been selected from which tracking of any object has to be started.
- From selected frame object to be tracked has been selected by repositioning the mask.
- For selected object its centroid position has been found out and from centroid information all the equation of time and measurement update have been calculated. For selected frame the actual position  $X$  and error  $P$  has been calculated.
- For all remaining frames following steps have been repeated.
  1. Background subtraction has been done to find out all the moving regions in the frame.
  2. From the found regions, region with the lowest distance from the region selected in previous frame has been selected.
  3. Selected region's centroid and other parameter have been used to calculate time and measurement update equations.
  4. Obtained state position values  $X$  has been stored in Array for every frame.
  5. Line joining each stored point has been drawn in every frame which shows the trajectory of the selected moving object.

### IV. RESULTS

The Tracking algorithm has been successfully applied to standard surveillance video of CAVIAR [6] and PETS [7] databases. Results shows that the algorithm has been able to track any single moving objects in indoor as well as in outdoor environments.

The algorithms have been implemented and tested on Matlab 2012a(64 bit) with operating system windows7. The system is having Pentium core i5 processor, 8 Gigabyte memory

CAVIAR dataset's video OneStopMoveNoEnter1cor.avi, with resolution 384 X 288 and 66 seconds in length has been reduced to 320 X 240 resolution and 28 seconds in length. Second video used for testing is PETS[7] dataset's outdoor video camera4.avi. The original video with 786 X 576 resolution of 107 seconds have been reduced to 320 X 240 resolution and 11 seconds in length.

Figure 2 shows output of frame no 50,300,650 for tracking using Kalman filter applied on standard surveillance video OneStopMoveNoEnter1cor.avi of CAVIAR[6] dataset. In the video two people(one man and woman) moves in the corridor. From the second frame a man has been selected for tracking and tracked in the subsequent frames. Black line has been drawn along the path on which man moves in the corridor.

Figure 3 shows output of frame no 50,200,250 for the Kalman filter applied on the PETS[7] dataset video camera4.avi. In the video few people and car are moving. At the initial stage white car has been selected for tracking. Black line has been drawn along the path of the car. The results shows the output of 50,200 and 250 frames.

Table 1 shows the time analysis for both the video. CAVIAR[6] dataset's video OneStopMoveNoEnter1cor.avi which have been reduced to 28 seconds have total 713 frames and system takes 74.47 seconds to process it. Third column shows the time analysis for PETS[7] dataset video Camera4.avi. The video have been reduced to 11 seconds with 343 frames. System takes 36.50 seconds to process the video.

TABLE I.  
Execution Time Analysis for Object Detection And Tracking

Parameter	Caviar Dataset Video OneStopMoveNoEnter1cor .avi	Pets Dataset Video Camera4.avi
Frame Height	240	240
Frame Width	320	320
Video Length in seconds	28	11
No of Frames	713	343
Total processing time in seconds	74.47	36.50



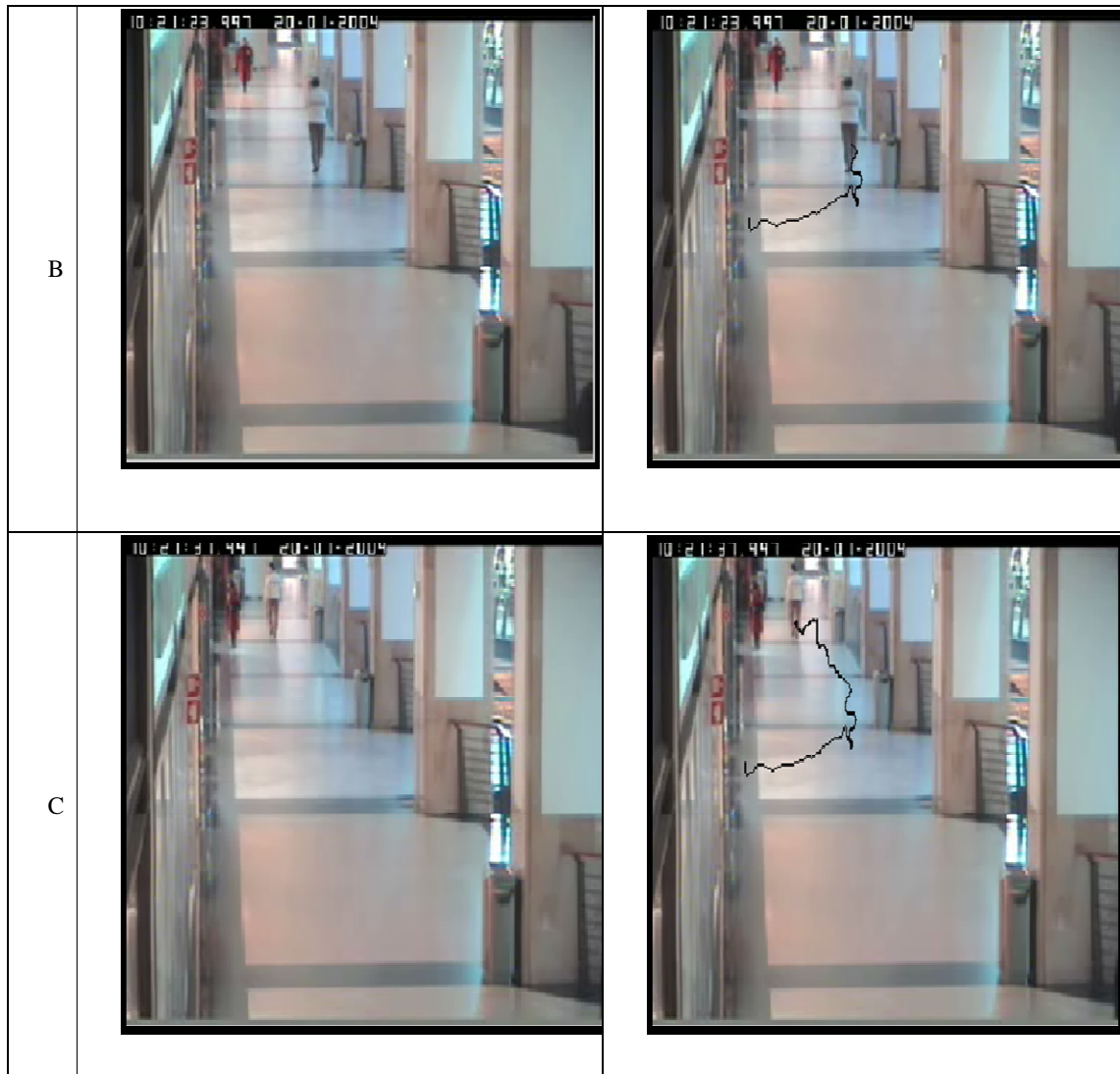


Fig 2.Object Tracking on CAVIAR[6] dataset video using Kalman filter. Left column indicates original frame and right column indicates tracked output[A]50<sup>th</sup> frame [B] 300<sup>th</sup> frame [C]650<sup>th</sup> frame

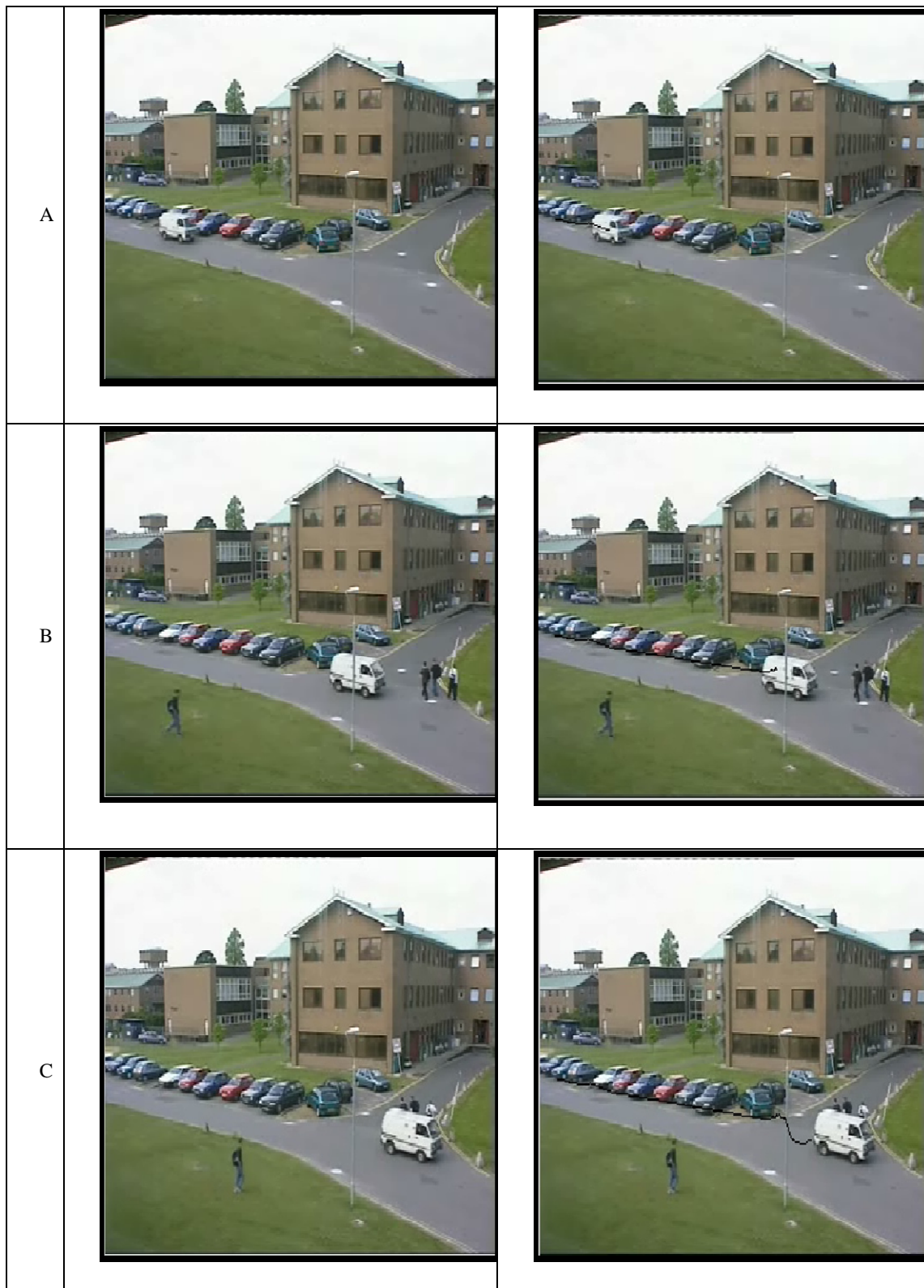


Fig. 3 Object Tracking on PETS[7] dataset video using Kalman filter. Left column indicates original frame and right column indicates tracked output[A]50<sup>th</sup> frame [B] 200<sup>th</sup> frame [C]250<sup>th</sup> frame

## V. CONCLUSION

In this work, a visual surveillance system with moving object detection and tracking capability has been presented. Object tracking of any single moving object has been successfully implemented on standard surveillance dataset of PETS [7] and CAVIAR [6] using Kalman filter. The system works on videos of indoor as well as outdoor environment taken using static camera under moderate to complex background condition. This implemented module can be applied to any computer vision application for moving object detection and tracking.

## VI. FUTURE WORK

This work can be extended to track multiple moving objects. The complete system with capabilities of detection and tracking can be used for applications domain like security, human computer interaction, Scene analysis and activity recognition, Event detection etc.

## REFERENCES

- [1] Kalman, R. E.: A new approach to linear filtering and prediction problems, Transaction of the ASME- Journal of Basic Engineering, 35–45, March 1960.
- [2] Watada JunzoMusa Zalili , Jain Lakhmi C., and Fulcher John., .," Human Tracking: A State-of-Art Survey.", KES 2010, Part II, LNAI 6277, pp. 454–463, 2010.
- [3] Welch, G. and Bishop, G.: An introduction to the Kalmann Filter, In University of North Carolina at Chapel Hill, Department of Computer Science. Tech. Rep. 95-041, available at <http://russelldavidson.arts.mcgill.ca/e761/kalman-intro.pdf>,2004.
- [4] Xu, Sheldon, and Anthony Chang. "Robust Object Tracking Using Kalman Filters with Dynamic Covariance." available at [http://www.cs.cornell.edu/Courses/cs4758/2011sp/final\\_projects/spring\\_2011/Xu\\_Chang.pdf](http://www.cs.cornell.edu/Courses/cs4758/2011sp/final_projects/spring_2011/Xu_Chang.pdf)
- [5] Yilmaz ,Alper, Omer Javed and Mubarak Shah. " Object tracking : A survey." *Acm Computing Surveys* (CSUR) 38.4 (2006):13.
- [6] <http://groups.inf.ed.ac.uk/vision/caviar/caviardata1/>
- [7] [http://www.hitech-projects.com/euprojects/cantata/datasets\\_cantata/dataset.htm](http://www.hitech-projects.com/euprojects/cantata/datasets_cantata/dataset.htm).