International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 6, June - 2013

Moving Object Tracking System In Video With Kalman Filter

Sumit Kumar Pal

M.Tech(ECE),Narula Institute of Technology

Kolkata -700109

Sohan Ghorai

Asst.Prof, ECE, Narula Institute of Technology

Kolkata- 700109

Abstract

Moving object tracking has a great importance in the area of computer vision. It has many real world applications like surveillance system, robotics, missile defence system, public security system and visual information processing. Many algorithms are in a line to solve that problem but performance sensitive to application under consideration and less immune to background noise. This project presents a real time tracking of a moving object in a video using Kalman filter. Kalman filter tracks an object by assuming the initial state and noise covariance. It provides an efficient method to calculate the state estimation process and make the minimum mean square error estimation. An experimental result from different moving object video samples shows a very good result. This filter is intended to be robust without being programmed with any environment specific rules.

1. Introduction

Object tracking on video sequence has many real world applications such as service robots, surveillance systems, public security systems, visual information processing and many more [1]. In defence applications, accurate tracking of missiles is very important for designing anti-missile solution. In automated Video Surveillance application, computer vision system is designed to monitor the movements in an area, identify the moving objects and report any doubtful situation. In some countries highway traffic is continuously monitored using cameras. Any vehicle that breaks the traffic rules or is involved in other illegal act can be tracked down easily if the surveillance system is supported by an object tracking system. Vehicle tracking is of major importance in applications of Intelligent Transportation Systems [2]. Object tracking is also extended for animation.

1.1 Background that motivated us

There are so many algorithms used for tracking purpose such as mean shift algorithm, camshift algorithm, optical flow, SURF etc. Each algorithm has strength in certain environment and weakness in others [3]. A summary of such algorithm are presented here.

Meanshift Algorithm: Meanshift algorithm is an efficient pattern-matching algorithm with no parameter estimation, and can be combined with other algorithms. It uses the kernel function histogram model of the target object. Meanshift is not sensitive for part of block rotation and deformation, but the track is easy to lapse when blocked a large area on the target [3, 4].
CamShift Algorithm: The Continuously Adaptive Mean Shift algorithm (CamShift) is a lightweight object tracking algorithm based on a one-dimensional hue histogram [9]. Originally designed for tracking faces or flesh tone, the algorithm computes the probability that any pixel is part of the tracked object as opposed to the background [3, 5]. The result of camshaft is not so good if the foreground is same colour as background or background's colour changes rapidly.
SURF: Speeded Up Robust Feature (SURF) is a feature detector and descriptor. It is based on sums of approximated 2D Harr wavelet responses and makes an efficient use of integral images [3]. The result of SURF is good even if the colour of background and foreground is same or background object's colour changes rapidly but the most crucial drawback is the computational time is high so it is incapable of tracking real-time object.
Optical Flow: Optical Flow tracking techniques allow the distinction between multiple objects and the background in a scene. This is computed based on the relative motion between an observer and the scene [6, 3].
A comparison chart of different has been given below [3].

"Table 1."

Algorithm	Dim light	bright light	lighting changes	moving object	stationary object	dropped frames	low resolution	similar background shapes	similar background colors
CamShift	bad	good	o.k.	good	good	o.k.	good	good	o.k.
SURF	good	good	good	N/A	good	good	bad	o.k.	good
Optical Flow	o.k.	o.k.	o.k.	good	bad	o.k.	bad	good	good

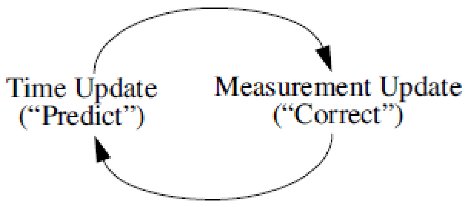
It is observed that most of the algorithm dependent on application environment and very less immune to noise. A good filtering algorithm can eliminate noise from the data and retain useful information. Kalman Filter, An optimal recursive data processing algorithm has been taken for this tracking problem. The Kalman filter not only works well in practice, but it is theoretically attractive because it can be shown that of all possible filters, it is the one that minimizes the variance of the estimation error.

2. Discrete Kalman Filter Theory

It is a recursive predictive filter that is based on the use of state space technique and recursive algorithm. It is called recursive because it does not require to store all the previous measurement and process the data optimally. Since the time of its introduction, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation [7]. The main advantages of kalman filter innovations are [8] (i) the variance of the Kalman filter innovations is smaller than the variance of the deterministic innovations. (ii) The computation time of Kalman filter is less. Generally it is two step process (i) Prediction (ii) Correction. First the state is predicted with the dynamic model then it is corrected with the observation model so that the error co-variance is minimized. The process is repeated with the each time with the step as the previous step as initial value.

2.1. Discrete Kalman Filter Algorithm

The Kalman filter estimates the process step as with the dynamic model and then take feedback in form of noisy measurements and update the estimates with the measurements. The equations for the Kalman filter fall into 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 estimates for the next time step. The measurement update equations are responsible for the feedback—i.e. for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equation is called predictor equation and the measurement update equation is called corrector equation. The complete algorithm is shown in Figure 1.



"Figure 1. The complete Kalman filter algorithm"

The time update equations are given as

$$\hat{X}_k^{-} = A\,\hat{X}_{k-1} + BU_k \text{ -------------------------(1)}$$

$$P_k^{-} = A\,P_{k-1}A^T + Q \text{ -------------------------(2)}$$

Where 'A' is state matrix, 'B' coverts control input and 'Q' is process noise covariance.

The measurement update equations are given as

$$K_k = P_k^{-}\,H^T\,(HP_k^{-}\,H^T + R)^{-1} \text{ -----------------(3)}$$

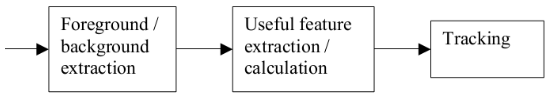
$$\hat{X}_k = \hat{X}_k^{-} + K_k\,(Z_k - H\,\hat{X}_k^{-}) \text{ -----------------(4)}$$

$$P_k = (1 - K_k H)\,P_k^{-} \text{ -------------------------------(5)}$$

The first task during the measurement update is to compute the Kaman gain, K_k . The next step is to actually measure the process to obtain Z_k , and then to generate a posterior state estimate by incorporating the measurement as in equation (4). The final step is to obtain a posterior error covariance with the equation (5,8). After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates.

3. Experiment Related Work Using Matlab

A pre-captured or real time video need to be processed before applying it in Kalman Filter. At first we need to read and represent a movie in a matrix form. This can be done in a matlab by using "mmreader" command. It constructs a multimedia reader object associated with video file. We can extract the information content in a video like height, width, number of frame associated in a video. After getting number of frames available in a video, we need to construct Matlab movie structure from the video frames. Matlab movie structure gives colour map information and 2D array of intensity values of 3 colours (RGB). Now we assumed that foreground contains the objects of interest / moving object. For removing less interest zones from the image, we need to extract background image by using mean filter.


Foreground / background extraction
Useful feature extraction / calculation
Tracking

For calculating the image containing only the background, a series of preceding images are averaged. For calculating the background image at the instant t,

$B(x,y) = \sum_{i=1}^{N} V(x,y,t-i)$

Where N is the number of preceding images taken for averaging. This averaging refers to averaging corresponding pixels in the given images. N would depend on the video speed (number of images per second in the video) and the amount of movement in the video. After calculating the background B(x,y) we can then subtract it from the image V(x,y,t) at time t =t and threshold it. Thus the foreground is

$| V(x,y,t) – B(x,y) | > Th$

Where 'Th' is threshold. Similarly we can also use median instead of mean in the above calculation of B(x,y). Once background has been extracted, now we use simple difference operation to get object of interest.

Now before calling Kalman Filter, image thresholding needed for removing artifacts and image smoothing. Since colour image is taken for processing, individual thresholding for Red, Green and Blue channel data needed. Then it removes object that have fewer than 300 pixel by using matlab command "bwareaopen". Thus it removes the noise artifacts.

After getting the number of frames from the video background is estimated and it is updated with the background updater. Here we track any moving object by initially tracking its centre and radius. Then a circle is drawn with the help of this centre and radius. Hence the model is independent of the size of the ball. So position and velocity will be the input for a kalman filter.

In our model of moving objects on 2D camera images, state is a 4-dimentional vector [x, y, dx, dy], where x and y represent the coordinates of the object's centre and dx and dy represent its velocity. The transition matrix is thus simply

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

Here B=[1 0 0 0] because object control affects only x-position co-ordinate.

For our model the measurement update equation $\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \begin{bmatrix} v\_k * randn \\ v\_k * randn \end{bmatrix}$

Therefore

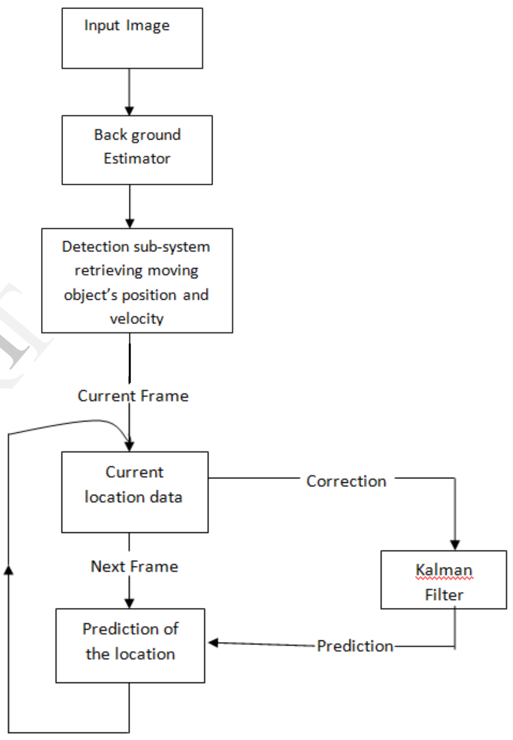
$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From one frame to other frame we see only change in the position (x,y) coordinate so measure vector is two dimensional vector. So

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

The measurement noise covariance matrix is calculated using the formula R= E[$V_k \;\; V_k^T$] where V_k is the standard deviation of the measurement noise.

A flow chart for object tracking problem formulation is given in figure 2.

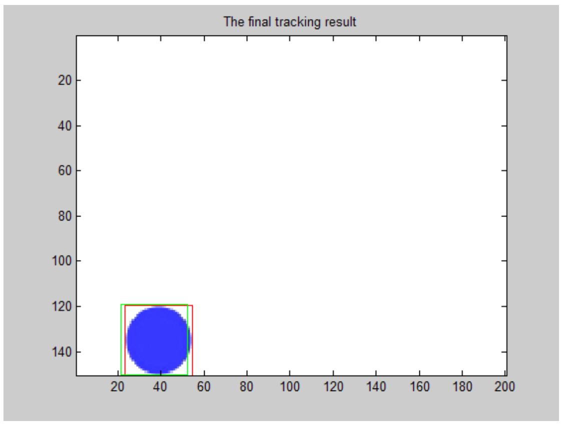


"Figure 2."

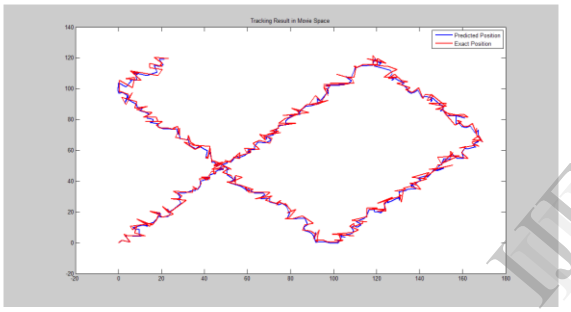
4. Experiments and Results

Tracking system hardware platform consist of two parts – PC and Camera. The first part is a PC with Intel Core2Duo 1.66 GHz processor and 2GB RAM with a web-camera mounted on it. Now we have collected small video samples containing a moving object to test our algorithm. Video has been taken from web-camera. Camera resolution is set to (320 x 240). Image acquisition frame rate is set to 30 frames per second. All video taken from camera is saved in Audio Video Intervened (AVI) format. Data rate is taken less than 1000kbps for faster decoding purpose.

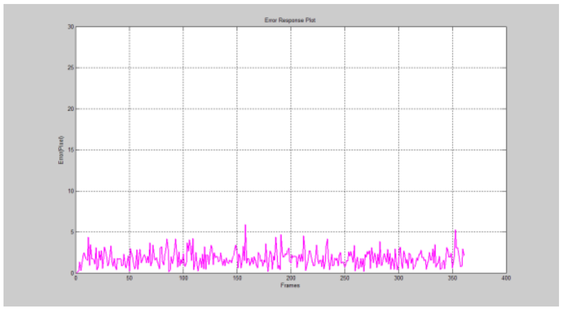
We have taken video named “SampleVideo.avi” that contains a random moving ball with some flicker noise. Video dimension is 200 x 150 pixel and frame rate is 30 fps. Tracking result and analysis report has been given below.



“Figure 3. Snap shot of a moving ball”



“Figure 4. The final tracking result in movie space of a moving ball”



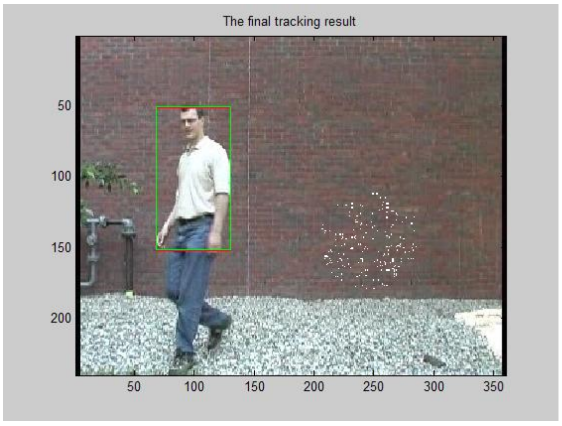
“Figure 5. The error response plot of a moving ball”

Kalman Filter Processing Time = 0.000058 seconds for 361 frames.

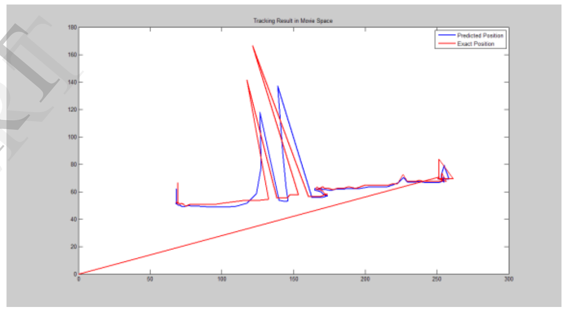
Standard deviation of errors = 0.9638

It can be seen from the result that, Kalman filter can successfully predict the path of that moving noise with very minimum noise of 5%.

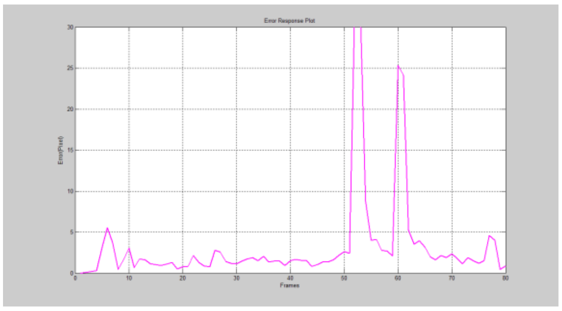
We have selected another video named “Man.avi” that contains a man walking on a road. Video dimension is 360 x 240 pixel and frame rate is 15 fps. Tracking result and analysis report has been given below.



“Figure 6. Snap shot of a man movement”



“Figure 7. The final tracking result in movie space of a man movement”



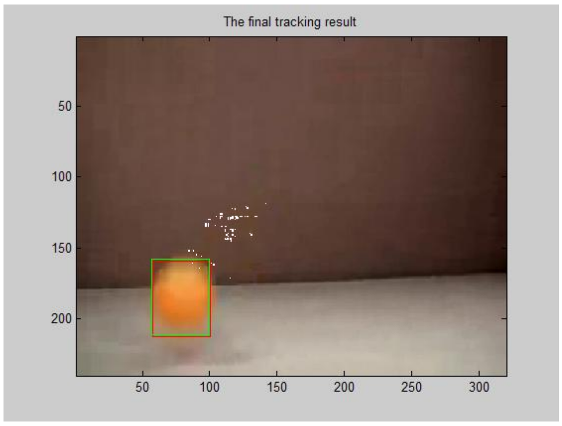
“Figure 8. The error response plot of a man movement”

Kalman Filter Processing Time = 0.000080 seconds for 80 frames.

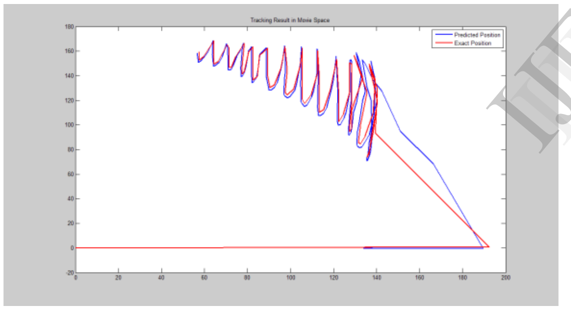
Standard deviation of errors = 6.1491

It is seen from the observation that since human body is an irregular object shape and irregular movement of hands and leg may create problem during Kalman prediction stage. Though it can track object successfully, but contains more than 30% error with high standard deviation.

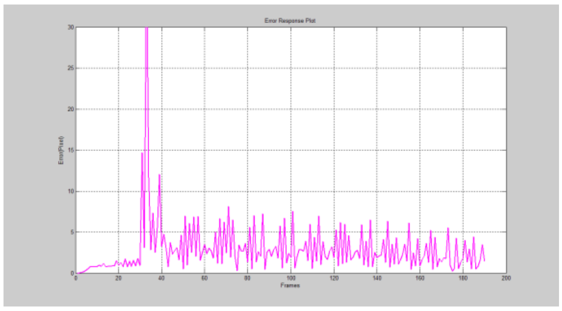
We have selected a real time video named "Bouncing ping pong ball.avi" that contains a ping pong ball free fall on a floor from human hand. Video dimension is 320 x 240 pixel and frame rate is 60 fps. Tracking result and analysis report has been given below.



"Figure 9. Snap shot of a bouncing ping pong ball"



"Figure 10. The final tracking result in movie space of a bouncing ping pong ball"



"Figure 11. The error response plot of a bouncing ping pong ball"

Kalman Filter Processing Time = 0.000114 seconds for 190 frames.
Standard deviation of errors = 3.3852

It is observed that, the video contains two motion transitions. A moving hand and a falling ball. At the time of transition error will be high. During falling stage, when ball bounce back after collision with floor, velocity changes rapidly and that causes a slight increase of error rate to near about 10%.

5. Conclusion

The program that was written in the scope of this paper intends to provide a basic understanding of the Kalman Filter. From the discussion, it can be seen that object tracking has many useful applications in the robotics and computer vision fields. Several researchers have explored and implemented different approaches for tracking. The success of a particular approach depends largely on the problem domain. In other words, a method that is successful in robot navigation may not be equally successful in automated surveillance. Further, there exists a cost/performance trade off. For real-time applications, we may need a fast high performance system on the other hand offline applications we may use a relatively cheap (and slower in performance). When real time response is required, the rate at which video is processed becomes very important. Real time systems should process the input at the rate equal to or faster than input data rate. If input is compressed domain data then the system needs to spend extra time to first uncompress data before processing them. This further reduces system performance. Thus having uncompressed input is desirable in real time system. In case of offline processing, cost becomes a critical factor. However we have successfully track the particular real time video and the result is quite good under assumptions.

6. References

[1] saeid bagheri-golzar et al., A New Method for Video Object Tracking , The Journal of Mathematics and Computer Science TJMCS Vol. 4 No.2 (2012) 122-128.
[2] Jianbo Shi and Carlo Tomasi, "Good Features to Track", IEEE Conference on Computer Vision and Pattern Recognition , 1994,pp. 593-600.
[3] Sheldon Xu and Anthony Chang, Robust Object Tracking Using Kalman Filter with Dynamic Covariance, Cornell University.
[4] Dorin Comaniciu, Peter Meer. "Mean Shift: A Robust Approach Toward Feature Space Analysis," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol 24(5),2002, pp:603-619.
[5] Zhang Hongzhi, Zhang Jinhuan, Yue Hui, Huang Shilin, "Object tracking algorithm based on

camshift", Computer Engineering and design,Vol 27(11), 2006, pp:2012-2014.
[6] Youssef ZINBI, Youssef CHAHIR, S."Moving object segmentation using optical flow with active contour model". IEEE Conference on ICTTA, 2008,pp. 1-5.
[7] Rachel Kleinbauer, Kalman Filtering Implementation with Matlab, Universitat of Stuttgart, Helsinki, Nov 2004.
[8] G. Welch and G. Bishop, An Introduction to Kalman Filter, proceedings of SIGGRAPH 2001,pp 19-24.
[9] J.G Allen et al., Object Tracking Using Camshift Algorithm Multiple Quantized Feature spaces, Conferences in Research and practice in Information Technology,vol. 36 ,204 pp 1-4