

Performance Analysis of Alpha Beta Filter, Kalman Filter and Meanshift for Object Tracking in Video Sequences

Dr. Ravi Kumar Jatoth

Department of Electronics and Communication, National Institute of Technology, Warangal, 506004, India
Email: jrk.nitw@gmail.com

Sanjana Gopisetty and Moiz Hussain

Department of Electronics and Communication, National Institute of Technology, Warangal, 506004, India
Email: {[g.sanjanarao](mailto:g.sanjanarao@gmail.com), [moizhussain.92](mailto:moizhussain.92@gmail.com)}@gmail.com

Abstract—Object Tracking is becoming increasingly important in areas of computer vision, surveillance, image processing and artificial intelligence. The advent of high powered computers and the increasing need of video analysis has generated a great deal of interest in object tracking algorithms and its applications. This said it becomes even more important to evaluate these algorithms to quantify their performance. In this paper, we have implemented three algorithms namely Alpha Beta filter, Kalman filter and Meanshift to track an object in a video sequence and compared their tracking performance based on various parameters in normal and noisy conditions. The proposed parameters employed are error plots in position and velocity of the object, Root mean square error, object tracking error, tracking rate and time taken to track the object. The goal is to illustrate practically the performance of each algorithm under such conditions quantitatively and identify the algorithm that performs the best.

Index Terms—Object tracking, Video tracking, Performance Analysis, Alpha Beta filter, Kalman filter and Meanshift.

I. INTRODUCTION

Object tracking is an important task within the field of computer vision [1]. From video surveillance to image processing, object tracking finds its applications in many areas. Object tracking which was once largely used for military and defense purposes is now rapidly finding its feet in commercial and industrial sector. For example, different sports like tennis and cricket have adopted the ball tracking technique (Hawk eye) in order to promote accurate decisions. Also, many CCTV cameras equipped with object tracking algorithms are being deployed on the roads in order to maintain and regulate traffic flow [1]. Another important area of application for object tracking is video surveillance where extraction of motion information is not only possible but also detection of any suspicious activities or unlikely events [1]. However,

tracking of objects is very complex in nature due to several problems such as presence of different noise in video, motion of objects, non-rigid or articulated nature of objects, partial and full object occlusion, changes in scene illumination and background [1].

To address these problems many performance metrics have been developed which take into account different parameters and conditions. Thus, evaluating the performance of tracking algorithms is becoming increasingly important and is an on-going research topic especially in video sequences. In this paper, we have focused our research in evaluating the above mentioned algorithms against their Root mean square error RMSE, Object tracking error OTE, Tracking detection rate, Peak signal to noise ratio PSNR, Boundary box error, error plots and time taken per frame in normal and noisy conditions. The algorithms have been simulated in MATLAB and results are analyzed and summarized.

The paper has been structured as follows: Section II gives the overview of the object tracking system, Section III presents the state modelling of the object, Section IV describes the different algorithms used, Section V discusses the evaluation measures and compares the object tracking methods and Section VI presents the concluding remarks of the paper.

II. OBJECT TRACKING SYSTEM IN VIDEO SEQUENCES

Tracking a moving object from a given video is done by extracting a sequence of images from the video at a given frame rate. After acquiring the sequence of images, detection of the object in every frame is done by distinguishing the foreground image from the background image using an object detection technique. Once the segmentation of the regions and detection of the object is achieved, we then employ tracking algorithms to track the objects from one frame to the next and generate its trajectory over time. An overview of this system described above is shown in Fig 1 [2]. After noting the tracked object's parameters, we proceed with evaluating each algorithm performance.

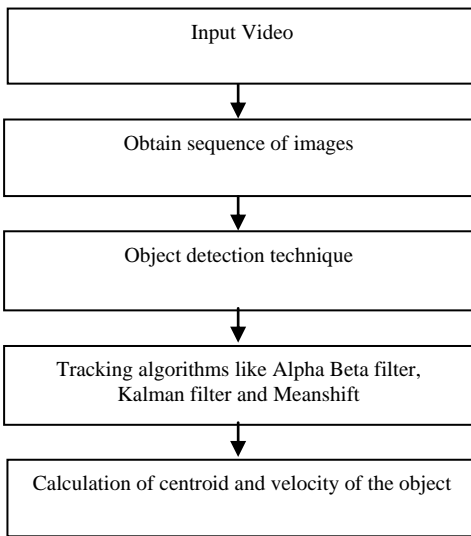


Fig 1. Object tracking system

III. MATHEMATICAL MODELING

Before we proceed with tracking an object, it is mandatory to model our system. This is done using the process state space equation given by (1). In this paper, we have modeled our system using the constant velocity model.

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ vx_{k+1} \\ vy_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ vx_k \\ vy_k \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & dt^2/2 & 0 \\ 0 & 0 & 0 & dt^2/2 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} \quad (1)$$

Where x_{k+1} and y_{k+1} are the x and y coordinates of center of object at the instant, x_k and y_k are the x and y coordinates of center of object at previous instant, vx_{k+1} and vy_{k+1} are the x and y component of velocity at the instant, vx_k and vy_k are the component of velocity at previous instant and a_x and a_y are accelerations of the object in x and y direction.

Since we have used the constant velocity model, the acceleration terms can be taken as process noise w_k . Thus, the above equation (1) can be written as (2),

$$x_{k+1} = Ax_k + w_k \quad (2)$$

Where A is called the state transition matrix and w_k is the Process noise given by $w_k \sim (0, Q)$ which is assumed as zero mean white Gaussian noise with process noise covariance Q .

Next step is to model the measurement space which is given by the measurement equation given below,

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} v_k * randn \\ v_k * randn \end{bmatrix} \quad (3)$$

which can be written as (4)

$$z_k = Hx_k + v_k \quad (4)$$

Where z_k is called the measurement vector, H is known as the measurement transition matrix and v_k is called the measurement noise given by $v_k \sim (0, R_k)$ which is assumed as zero mean white Gaussian noise with measurement noise covariance R_k .

IV. OBJECT TRACKING ALGORITHMS

A. Alpha Beta filter

Alpha Beta tracker is similar to Kalman filter but a simplified form of it. Unlike the Kalman filter it uses a simple static system model. It is generally used for steady state stationary condition in which tracking is characterized by constant velocity of the object and constant measurement noise covariance.

It is implemented using two steps - i) Prediction and ii) Correction. The former step includes the prediction of the current set of object parameters that is the position and velocity estimate. An error is calculated based on the true and predicted values of object parameters. The error in position and velocity predicted is corrected using the static Alpha and Beta gains respectively. Alpha and beta gains are initialized in the beginning of the algorithm.

The underlying alpha beta model described above is represented by mathematical equations from (5) to (8) [3][4][5].

Prediction:

$$x_k = x_{k-1} + Tv_{k-1} \quad (5)$$

$$v_k = v_{k-1} \quad (6)$$

Correction:

$$x_k = x_k + \alpha(x_k - x_k^-) \quad (7)$$

$$v_k = v_k + (\beta/T)(x_k - x_k^-) \quad (8)$$

Where x_k is the predicted position, x_{k-1} is the previous position; v_k is the predicted velocity of object and v_{k-1} is the previous velocity of object; x_k^- is the true position of the object, $x_k - x_k^-$ is the amount of error in prediction, α and β are called the Alpha and Beta gains respectively and T is the sampling period.

B. Kalman filter

Kalman filter is a recursive filter that provides efficient tracking estimations of the object even in presence of Gaussian noises [6]. It uses a detailed dynamic system model compared to that of Alpha Beta filter. Kalman filter because of its small computation requirement and

efficient recursive property is popularly used as an optimum tracking algorithm in various fields [6]. The algorithm is implemented using two steps – i) Prediction ii) Correction.

Initially, state transition matrix A , Process noise covariance Q , measurement noise covariance R and measurement transition matrix H are initialized. The next step involves the prediction of the present state of the object and error covariance described by equations from (9) to (11) [6][7][8][9]. This predicted state estimate is called the priori estimate. The state of the object is represented by 4x1 state matrix x given by $[x \ y \ v_x \ v_y]$ where x and y represent the coordinates of the object and v_x and v_y represent the object's velocity in x and y direction respectively.

Prediction:

$$x_k^- = Ax_{k-1} + w_{k-1} \quad (9)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (10)$$

$$p(w) \sim N(0, Q) \quad (11)$$

Where, A is called the state transition matrix, x_{k-1} is the previous state, x_k^- is the predicted state, P_k^- is the predicted error covariance, P_{k-1} is the previous error covariance and Q is called the process noise covariance.

Next, the Kalman filter uses the state prediction x_k^- and error covariance at k instant to correct its prediction by using Kalman gain K . This corrected estimate is called posteriori estimate which is described by equations from (12) to (15) [6][7][8][9]. The equations in-cooperate a new measurement into the priori estimate to obtain an improved posteriori estimate. z_k is the true state of the object's x and y coordinates and H is the measurement transition matrix which maps the observed space from true space.

Correction:

$$K = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (12)$$

$$x_k = x_k^- + K(z_k - Hx_k^-) \quad (13)$$

$$P_k = (I - KH)P_k^- \quad (14)$$

$$p(v) \sim N(0, R) \quad (15)$$

Where K is the Kalman gain, P_k^- is the predicted error covariance, H is the measurement transition matrix, R is the measurement noise covariance, x_k is the corrected state, z_k is the measurement matrix and P_k is the corrected error covariance.

From the above equations it is evident that the Kalman filter takes into the dynamics of the environment by considering the noise covariance.

C. Meanshift algorithm

Mean shift is a nonparametric iterative algorithm [10][11]. It iteratively shifts the data point to the average of data points in its neighborhood. For each data point, mean shift defines a window around it and computes the mean of data point. Then it shifts the center of window to the mean and repeats the algorithm till it converges [10].

Given n data points $x_i, i=1,2,\dots,n$ on a d dimensional space R^d , the multivariate kernel density estimate obtained with Kernel $K(x)$ and window radius h is given by (16).

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (16)$$

The radially symmetric kernel is defined by (17)

$$K(x) = c_k k(\|x\|^2) \quad (17)$$

Where c_k is the normalization constant. Taking the gradient of the density estimator and further algebraic manipulation yields,

$$\nabla f(x) = \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \right] * \left[\frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x \right] \quad (18)$$

Where $g(x) = -k'(x)$ denotes the derivative of the selected kernel profile. The first term is proportional to the density estimate at x (kernel $G = c_g g(\|x\|^2)$). The second term is the mean shift vector M which points to the direction of maximum increase in density. The mean shift algorithm is summarized as follows:

- i) Compute mean shift vector $M(x_i^t)$.
- ii) Translate density estimation window using $x_i^{t+1} = x_i^t + M(x_i^t)$
- iii) Iterate the above steps i) and ii) until they convergence i.e. $\nabla f(x_i) = 0$

V. EXPERIMENTAL SIMULATION AND RESULTS

A. Video Specifications

Video length: 2.4 sec
Image dimension: 1270*720 pixels
Image type: RGB color space
Height: 720 pixels
Width: 1280 pixels
No of frames: 73
No of frames with object: 36

B. Experimental Results

In this paper, we have analyzed the translational motion of the object. In doing so, we can observe the object parameters in real and relate these parameters to the tracked object's estimated parameters with ease. Comparison and evaluation of the above mentioned tracking algorithms are done using parameters like RMSE, OTE, error plots and time analysis. In addition, Gaussian noise is added (which represents rainy weather condition) to the images and object tracking algorithms are applied under these conditions to compare their performance. Simulation of the video sequences under normal and noisy conditions for the three algorithms is shown below,



Fig 2. Alpha Beta tracking under normal conditions

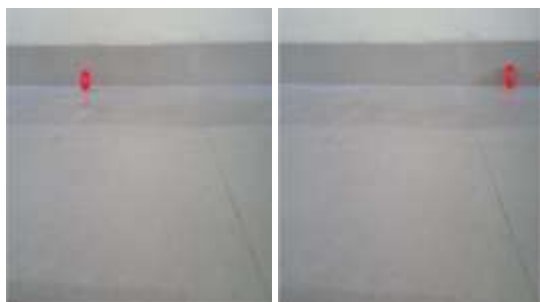


Fig 3. Kalman tracking under normal conditions



Fig 4. Meanshift tracking under normal conditions



Fig 5: Alpha Beta tracking under noisy conditions

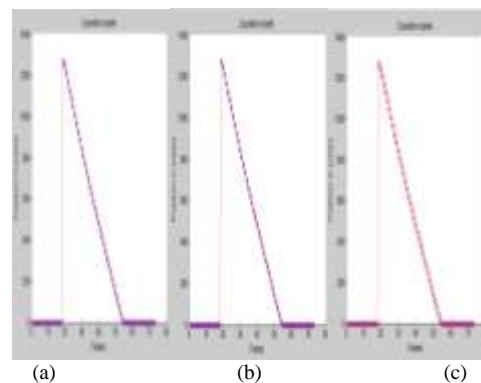


Fig 6: Kalman tracking under noisy conditions

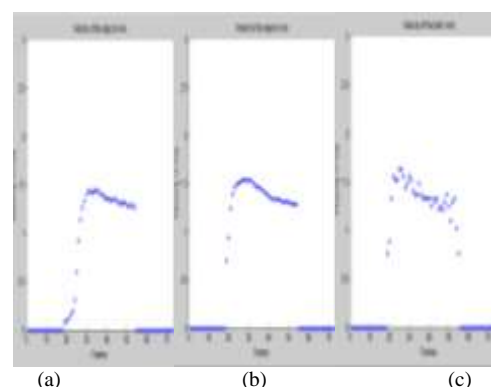


Fig 7: Meanshift tracking under noisy conditions

The position estimate is plotted in Fig 8 under normal conditions. Red indicates the object tracked and Blue indicates the actual position of the object.

Fig 8. (a) $\alpha\beta$ filter (b) Kalman filter (c) Meanshift position estimate

The velocity estimate for the tracking algorithms is plotted in Fig 9 under normal conditions. The true average velocity of the object is 1.375m/s. The estimated average velocity for $\alpha\beta$ filter is 1.117m/s, Kalman filter is 1.371m/s and Meanshift is 1.362m/s.

Fig 9. (a) $\alpha\beta$ filter (b) Kalman filter (c) Meanshift velocity estimate.

Now that we know the true and tracked object's parameters, we can evaluate their performance. In this paper we have compared the tracking algorithms using the following set parameters. Although there are many metrics in evaluating the tracking performance, we have limited to the following seven parameters as they can implemented with ease.

B.A. Absolute error

Absolute error is the magnitude of difference between the true value and the tracked value of the object. It is given by,

$$\mathcal{E} = |O_{true} - O_{tracked}| \quad (19)$$

Where o_{true} is the true value of object's parameters and $o_{tracked}$ is the tracked value of the object's parameters.

The absolute error in position and velocity is shown in Fig 11 and Fig 12 respectively for the three tracking algorithms. We can notice that Kalman filter has the least error in estimating the object's position and velocity.

B.B. Root Mean Square Error

RMSE measures the difference between values predicted and the values actually observed. It is given by,

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (o_{predicted} - o_{observed})^2}{n}} \quad (20)$$

Where $o_{predicted}$ is the tracked estimation of the object's parameters, $o_{observed}$ is the true estimation of the object and n is the number of frames with the object present.

The RMSE in position and velocity is tabulated in Table I and II under normal and noisy conditions. We can notice that Kalman filter has the lowest RMSE error, followed by Mean shift and Alpha Beta filter which has the highest RMSE error in normal and noisy conditions.

B.C. OTE error

Object tracking error (OTE) is the average discrepancy in the centroid of the tracked object from its true value. It is given by [12],

$$OTE = \sqrt{\frac{\sum_{i=1}^n (x_{true} - x_{tracked})^2 + (y_{true} - y_{tracked})^2}{n}} \quad (21)$$

Where x_{true} and y_{true} are the actual x and y coordinate of the object, $x_{tracked}$ and $y_{tracked}$ are the tracked x and y coordinate of the object.

Since the errors in determining position is least for Kalman followed by Mean shift and Alpha Beta filter. The OTE for Kalman is expected to be least among the other two tracking algorithms in noisy and normal conditions. The values are summarized in Table I and II and OTE per frame is plotted in Fig 13.

B.D. Tracking detection rate

Tracking detection rate refers to the amount of success in detecting an object [14].

$$TDR = \frac{\text{no of frames object detected}}{\text{no of frames with object present}} * 100 \quad (22)$$

From the Table I and II, it is evident that all the three algorithms have good tracking rate in normal conditions as opposed to noisy conditions where there are missed detections.

B.E. Bounding box error

Bounding box error refers to the overlap area error of the tracked object to the actual object [13]. If BBE is 1, then the tracking is accurate to the true object location and size and 0 indicates the worst performance of the tracking algorithm i.e. no object tracked.

$$BBE = \frac{\text{Area}(A \cup T)}{\text{Area}(A \cap T)} \quad (23)$$

Where A is the true area of the object and T is the area of the object tracked. Fig 10 shows an object outlined by green and object tracked outlined by red. The shaded areas represent intersection and union of A and T respectively.

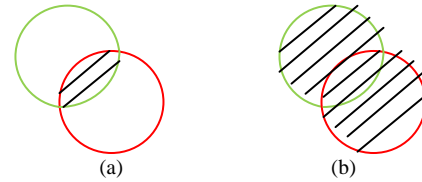


Fig 10. (a) $A \cap T$ (b) $A \cup T$

From the Fig 14, we notice that Kalman Filter performs better in detecting the object and has less BBE as compared to the other algorithms.

B.F. Noise performance

To evaluate the performance in noisy conditions, we use Peak signal to noise ratio which gives ratio between the maximum possible power of a signal and the power of corrupting noise.

Alpha Beta requires a high PSNR of about 19 db for tracking the object in noisy conditions reasonable well whereas Kalman and Mean shift both perform even better at that value. The min PSNR required for Mean shift is 16 db and for Kalman is 15 db. Thus, we conclude that Kalman filter performs well in low values of PSNR. The performance for a given PSNR value is summarized in Table II.

B.G. Time analysis

By evaluating the time taken to track the object we can determine which algorithm can work faster. Fig 15 shows the time taken to track the object per frame. We notice that Kalman takes overall less time per frame to track the

object location compared to Meanshift and Alpha Beta filter.

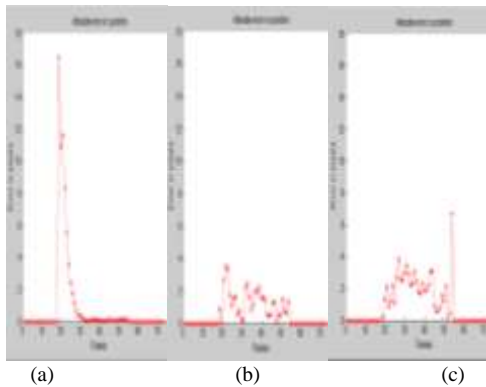


Fig 11. (a) $\alpha\beta$ filter (b) Kalman filter (c) Meanshift Absolute error in position per frame.

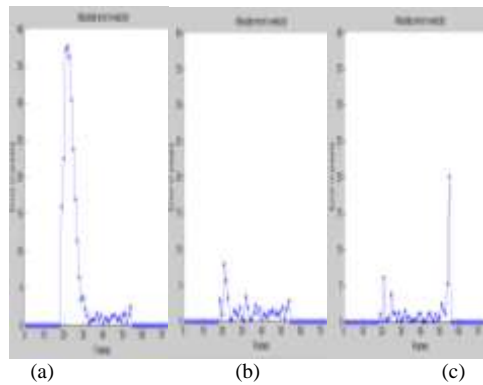


Fig 12. (a) $\alpha\beta$ filter (b) Kalman filter (c) Meanshift Absolute error in velocity per frame.

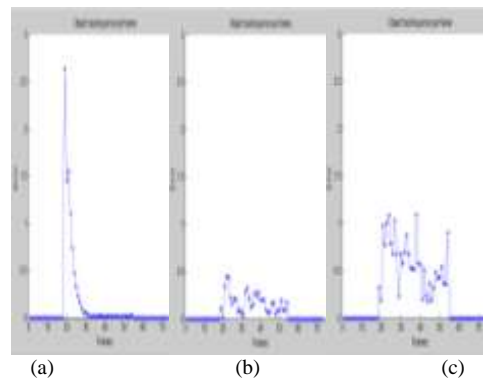


Fig 13. (a) $\alpha\beta$ filter (b) Kalman filter (c) Meanshift OTE per frame.

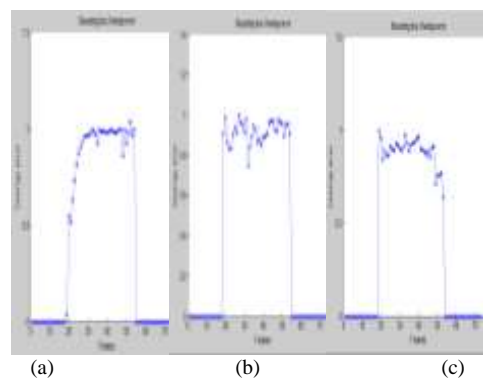


Fig 14. (a) $\alpha\beta$ filter (b) Kalman filter (c) Meanshift Bounding Box error.

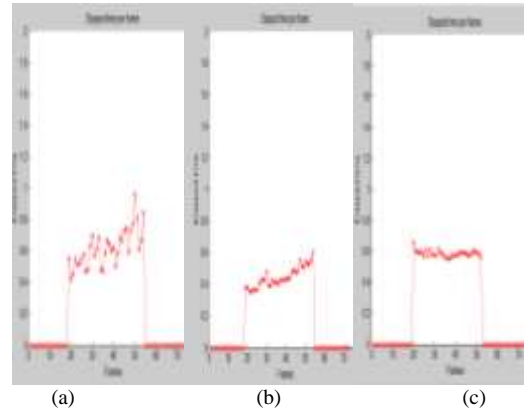


Fig 15. (a) $\alpha\beta$ filter (b) Kalman filter (c) Meanshift Time taken per frame.

TABLE 1. Under Normal Conditions

Algorithm	Performance parameters			
	RMSE Position (cm)	RMSE Velocity (cm/s)	OTE (cm)	TDR
$\alpha\beta$ Filter	0.5473	1.7803	0.7532	35/36=97.22%
Kalman filter	0.2002	0.2867	0.1642	36/36=100%
Meanshift	0.3012	0.4764	0.6748	36/36=100%

Table 2. Under Noisy Conditions

Algorithm	Performance parameters				
	RMSE Position (cm)	RMSE Velocity (cm/s)	OTE (cm)	TDR	PSNR (db)
$\alpha\beta$ Filter	23.971	26.971	33.346	33/36=91.66%	18.39
Kalman filter	0.3012	0.4013	0.1721	36/36=100%	18.39
Meanshift	0.5059	0.6351	1.0380	36/36=100%	18.39
$\alpha\beta$ Filter	0.5729	1.9203	1.7969	35/36=97.22%	20.15
Kalman filter	0.2012	0.2967	0.1670	36/36=100%	20.15
Meanshift	0.4920	0.6348	0.7449	36/36=100%	20.15

VI. CONCLUSIONS

Comparing and evaluating the simulation results for Alpha beta filter, Kalman and Meanshift the following conclusions can be drawn regarding their performances. Firstly, Kalman filter has the least error in calculation of the object trajectory compared to Meanshift and Alpha beta filter. Thus, as expected the OTE and RMSE errors for Kalman are quite low. Secondly, Kalman filter can precisely detect and track the ball and hence provides the least boundary box error compared to the other algorithms. Lastly, Alpha Beta performs the worst in noisy conditions and needs a high value of PSNR to work reasonably well. Overall, Kalman filter is preferable over Meanshift and Alpha beta filter.

REFERENCES

- [1] Alper Yilmaz, Omar Javed, Mubarak Shah "Object Tracking: A Survey" in ACM Computing Survey Volume 38, Dec 2006, pg-2
- [2] Abhishek Kumar Chauhan, Deep Kumar "Study of Moving Object Detection and Tracking of Video Surveillance" in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013.
- [3] D.M Akbar Hussain, David Hicks, Daniel Orti Arroyo "Case Study: Kalman and Alpha Beta Computation under High Correlation" in Proceedings of International Multi-Conference of engineers and computer scientists Vol I, March 2008.
- [4] M. Munu Harrison and M.S. Woolfson, Comparison of the Kalman and α - β Filters for the Tracking of Targets Using Phased Array Radar, University of Nottingham, U.K, Vol 4, 2012
- [5] Jae-Chern Yoo, Young-Soo Kim. Alpha-beta-tracking index tracking filter, In POSTECH Information Research Lab, Pohang University of Science and Technology, South Korea, Vol 5 2003
- [6] Ramsey Faragher, Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation, In IEEE Signal Processing Magazine, September 2012.
- [7] Greg Welch, George Bishop "An Introduction to the Kalman filter" Sep 1997, pg 1 -6.
- [8] Hitesh A Patel, Darshak G Thakore "Moving Object Tracking Using Kalman Filter" IJCSMC, Vol. 2, Issue. 4, April 2013, pg.326 – 332
- [9] Zhaoxia Fu, Yan Han. Centroid weighted Kalman filter for visual object tracking, In Information and Communication Engineering Institute, North University of China, Taiyuan, Elviewer 2012.
- [10] Dorin Comaniciu, Peter Meter "Meanshift: A robust approach towards feature space analysis" IEEE transactions on pattern analysis and machine intelligence of 24, May 2002.
- [11] Y. Cheng. Mean shift, mode seeking, and clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence, 17 (8): pg-790-799, 1998.
- [12] Faisal Bashir, Fatih Porikli "Performance evaluation of object detection and Tracking systems" in Mitsubishi Electric Research Laboratories June 2006.
- [13] Fei Yin Dimitrios Makris, Sergio Velastin " Performance evaluation of object tracking algorithms" in Digital Imaging Research center.
- [14] V Purandhar Reddy, K Thirumala Redd, YB. Dawood "Performance evaluation of object tracking technique based on position vectors", International journal of image processing Vol 7, Issue 2 2013.

Institute of Technology Warangal, India in 2014. He is currently working as Lecturer in National Institute of Technology Warangal, India. His current research areas include Digital Signal Processing, Tracking Algorithms and Process controller design.



Sanjana Gopisetty, Female, has completed her bachelor's degree in Electronics and Communication Engineering at the National Institute of Technology Warangal, Andhra Pradesh, India in 2014.



Moiz Hussain, Male, has completed his bachelor's degree in Electronics and Communication Engineering at the National Institute of Technology Warangal, Andhra Pradesh, India in 2014.

Authors' Profiles



Dr. Ravi Kumar Jatoth, Male, has received his B.E degree in Electronics and Communications Engineering from Osmania University, Hyderabad, India in 2003 and M.Tech in Instrumentation and Control Systems from Jawaharlal Nehru Technological University, Hyderabad, India in 2005. He has also received his PHD degree from National