Indian Journal of Science and Technology, *Vol 9(30), DOI: 10.17485/ijst/2016/v9i30/99017, August 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Six Object Tracking Algorithms: A Comparative Study

Vaibhav Kumar Agarwal[1*], N. Sivakumaran[1] and V. P. S. Naidu[2]

[1]Department of Instrumentation and Control Engineering, National Institute of Technology, Tiruchirappalli - 620015, Tamil Nadu, India; Vaibhav.agarwal100790@gmail.com
[2]Flight Mechanics and Control Division, MSDF Lab, CSIR-NAL, Bangalore - 560017, Karnataka, India; nsk@nitt.edu, vpsnaidu@gmail.com

Abstract

Objectives: To compare five different objects tracking algorithms performance wise with the proposed algorithm and to find out the best one among them for tracking of an object in occlusion and Background clutter condition i.e. when background contains target features. **Methods/Analysis:** In the proposed algorithm, the object's position is obtained by Mean Shift tracking and then the prediction of object's position is done through Kalman Filter. The Proposed method has Kalman Filter which consists of an adaptive system matrix and adaptive process error covariance and measurement error covariance matrices respectively. Adaptive system matrix of Kalman Filter is getting updated online depending on the quality of observation by Mean Shift algorithm and adaptive process and measurement noise covariance matrices are getting updated according to the variation in Bhattacharya Coefficient respectively. **Findings:** Proposed algorithm has a maximum value (0.2177 and 0.4821) of tracking efficiency metric i.e. Bhattacharya Coefficient in both video dataset among all the algorithms and it is taking less execution time (0.002761 sec and 0.005431 sec) than other Kalman filter based tracking algorithms. **Applications/Improvements:** Proposed algorithm is able to track the target in occlusion and background variation condition with more efficiency and less execution time than rest of the algorithms.

Keywords: Adaptive, Covariance, Kalman Filter, Mean Shift, Occlusion, Tracking

1. Introduction

Tracking of an object in a playing video has become an important area of research in computer vision and Image Processing. Tracking of the object is done in many applications such as tracking of a fleeing criminal, surveillance, smart rooms, intelligent robotics and human-computer interaction[1,2]. Many tracking algorithms are available for tracking of an object such as Cam Shift, Mean Shift[3] etc. and all have some limitations[4]. Mean Shift (MS) is an iterative non-parametric density estimation method which calculates the nearest mode of density[5,6] and is efficient in target tracking under normal conditions[7] but fails in tracking under a background clutter[8,9], noisy environment, partial or full occlusion, a sudden change in speed of object etc.[10,11]. Kalman Filter is a state estimate algorithm[12] which can be used to eliminate partial and full occlusion and sudden velocity change problem[10,11]. Discussed algorithms take the use of both Mean Shift and Kalman Filter and manipulate the data according to the need for efficient tracking of the object. Firstly, object's position is predicted by Kalman Filter (KF) and that position is used as an observation for Mean Shift (MS) algorithm then the distance between the predicted position and Mean Shift Position is calculated and if the distance is less than threshold then Mean Shift value is used in Update Equations of Kalman filter for getting exact location of object otherwise predicted value of Kalman filter is used. Since no process in this world is linear so, extended or unscented Kalman Filters can be used for non-linear motion tracking or for getting better performance[13].

In this paper, five different object tracking algorithms are implemented and compared performance wise with the new approach for adaptive Kalman Filter which is proposed to eliminate the drawbacks of Mean Shift algorithm

** Author for correspondence*

2. Six Different Tracking Algorithms

Many object tracking algorithms are available[1,2] and all have one or another limitations[3,4]. For e.g. Mean Shift (MS) works on the color property of target and Kalman Filter (KF) works on equations of motion. Mean Shift (MS) is a non-parametric density estimator that iteratively computes the nearest mode (maxima) of a sample distribution[6]. In MS tracking algorithm, to represent the target, a color histogram is used because of its capability to remove scaling, rotation and partial occlusion. For determining the similarity between the target model and the candidate model of the tracking window, Bhattacharya coefficient is used[6,7] or Bhattacharya Coefficient can be defined as the metric for measuring tracking efficiency. In MS algorithm, background information is also included in the detected target region. If the connection between target and background is high, accuracy in the detection of an object and in its localization will be decreased for Mean Shift algorithm[8,9]. To decrease the background information or connection, Mean Shift with CBWH (MS-CBWH) is proposed for removing error due to background[9].

Mean Shift with CBWH and Kalman Filter (MS-CBWH-K) is a powerful scheme for tracking of an object under occlusion[14] but due to fixed parameters, tuning task of this KF is tedious. Mean Shift with CBWH and Kalman Filter with Adaptive System Matrix (MS-CBWH-AKF) is a dynamic scheme for the up gradation of system matrix of Kalman Filter online depending on the quality of tracking due to which partial and full occlusion problem can be eliminated very easily[15,18] but Q (Process error covariance matrix) and R (Measurement error covariance matrix) tuning of this Kalman Filter is difficult. Mean Shift with CBWH and Kalman Filter with Adaptive Q and R (MS-CBWH-AKQR) is a novel technique for tracking of the target under occlusion and with theauto tuning of Q and R parameters[16] but it has constant system matrix.

2.1 Mean Shift with Corrected Background Weighted Histogram and Kalman Filter with Adaptive F, Q and R (MS-CBWH-AKFQR)

This is an approach in which all the parameters of KF are getting updated online depending on Bhattacharya Coefficient and MS observation respectively. The tedious procedure in Kalman Filtering is the tuning of its process error covariance (Q) and measurement error covariance (R) matrices so that it can work efficiently. In Kalman Filter, measurement error covariance matrix R(k) and Kalman gain K are in inverse ratio i.e. whenever the value of R(k) is decreased the value from the measurement or from the sensor is weighted more and this value is trusted more and the predicted value is trusted less because the Kalman gain weights the residual more heavily[16,17]. On the other hand, when the value of Process Error Covariance matrix Q(k) is less, then the measurement is trusted less and the predicted value is trusted more and more.

In this paper, when Kalman Filter is applied with Mean shift then with the increasing value of Q(k), tracking of an object is more dependent on the Mean Shift algorithm (MS) or color property of the object and less dependent on the predicted value of Kalman Filter or equations of motion. Similarly, when the value of R(k) is increased, tracking of an object is more dependent on the predicted value of Kalman Filter or equations of motion and less dependent on the Mean Shift algorithm or color property of the object. Therefore, optimal results can be obtained if we will be able to decide which one to trust. In this work, the so-called adaptive KF allows the estimated parameters Q(k) and R(k) of KF to adjust automatically according to the Bhattacharya coefficient of MS object tracking because Bhattacharya Coefficient is the measure of tracking of an object or Bhattacharyya coefficient evaluates the similarity of the target and candidate models.

Since when the object is occluded by another object then Bhattacharya coefficient will decrease accordingly. Hence, the threshold can be set for determining occlusion i.e. when Bhattacharya Coefficient is below the threshold then an occlusion is there and the algorithm should be Kalman Predictive property dominated and when it is above the threshold then it should be dominated by Mean Shift algorithm. For the up gradation of F(k) (system matrix), if object's motion is not constant then the observation obtained by Mean Shift algorithm is considered otherwise it is not considered.

In this paper, an adaptive Kalman Filter is designed which has variable F, Q, and R matrices and the parameters are varying according to the error between MS observation and KF prediction and Bhattacharya coefficient respectively.

In this method, system matrix is given as:

$$F_{k+1|k} = \begin{bmatrix} 1 & 0 & dX_{k+1|k} & 0 \\ 0 & 1 & 0 & dY_{k+1|k} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Where $dX_{k+1|k}$ and $dY_{k+1|k}$ are the distance moved by center of tracking window and are getting updated as follows:

$$dX_{k+1|k} = (1 - a_k)dX_{k|k-1} + a_k(\hat{X}_k - \hat{X}_{k-1}) \quad (2)$$

$$dY_{k+1|k} = (1 - a_k)dY_{k|k-1} + a_k(\hat{Y}_k - \hat{Y}_{k-1}) \quad (3)$$

Where $\hat{X}_k$ and $\hat{Y}_k$ are the estimated (Mean Shift Values) values of center x and y respectively.

And

$$ak(s) = f(d(s)) \quad (4)$$

Where d(s)is given by

$$d(s) = \sqrt{1 - \eta(k)} \quad (5)$$

Where

$\eta(k)$ is the Bhattacharya Coefficient

d(s)is the distance between the model and the candidate histogram at the position s.

f(s)is a decreasing function given by

$$f(s) = 1-s \quad (6)$$

The estimate $\hat{X}_{k+1|k}$ contributes to the updates of the displacement $d_{k+1|k}$ only when the current estimate resembles the source object model, which is when $a(s) \rightarrow 1$. On the other hand ($a(s) \rightarrow 0$), the displacements included in the state matrix remains nearly unchanged, as they were in step k, considering that the object is occluded. This process has the advantage that matrix $F_{k+1|k}$ incorporating information on the object movement can be updated by the tracking algorithm.

$$Q(k) = \begin{bmatrix} \lambda_1(k) & 0 & 0.5\lambda_1^2(k) & 0 \\ 0 & \lambda_1(k) & 0 & 0.5\lambda_1^2(k) \\ 0.5\lambda_1^2(k) & 0 & \lambda_1(k) & 0 \\ 0 & 0.5\lambda_1^2(k) & 0 & \lambda_1(k) \end{bmatrix} \quad (7)$$

$$R(k) = \begin{bmatrix} \lambda_2(k) & 0 \\ 0 & \lambda_2(k) \end{bmatrix} \quad (8)$$

Where

$$\lambda_1(k) = (1 - \rho)\hat{\lambda}_1(k) + \rho\lambda_1(k-1) \quad (9)$$

$$\lambda_2(k) = (1 - \rho)\hat{\lambda}_2(k) + \rho\lambda_2(k-1) \quad (10)$$

And

$\hat{\lambda}_1(k) = \eta(k)$ If $\eta(k) >= B_T$
Otherwise $\hat{\lambda}_1(k) = 0$ (11)

$\hat{\lambda}_2(k) = 1 - \eta(k)$ If $\eta(k) >= B_T$
Otherwise $\hat{\lambda}_2(k) = \psi$ (12)

Where

$\eta(k)$is Bhattacharya coefficient

B_T is a threshold value

ψ is a large constant

So the posterior estimate of KF approximates to its predicted value, and $\rho \in [0, 1]$ is the forgetting factor. The lower ρ is, the faster is the update of $\lambda_1(k)$ and $\lambda_2(k)$.

- Measurement matrix H is given as:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (13)$$

- State covariance matrix is given as:

$$P_{\text{int}} = 10 * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

- System matrix F is given as:

$$F_{\text{int}} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

- Initial state vector of an object is given as:

$$X_{int} = \begin{bmatrix} x_C(0) \\ y_C(0) \\ 0 \\ 0 \end{bmatrix} \quad (16)$$

2.2 Algorithm for Kalman Filter Implementation

Step 1: Initialize the Kalman Filter parameters like X_{int}, P_{int}, Q, R and F.

Step 2: Predict the initial position of the center of an object using Kalman filter prediction equation or take it as the initial center i.e. center of the object in the starting frame.

- $\tilde{X}_{k|k-1} = F_k \hat{X}_{k-1|k-1}$
- $\tilde{P}_{k|k-1} = F_k \hat{P}_{k-1|k-1} F_k^T + Q_k$

Step 3: Use this predicted location of center as an input to Mean Shift Algorithm.

Step 4: Calculate the distance between predicted object center and measured object center from Mean Shift for every iteration, if the distance is more than the threshold then consider center as Kalman predicted center otherwise consider center as Mean Shift predicted center.

$$d(k) = \sqrt{(\hat{x}(k) - x_C(k))^2 + (\hat{y}(k) - y_C(k))^2}$$

Step 5: Resulting center of **Step 4** should be given to Kalman Filter update equations to find the true value of center

- $K_k = \tilde{P}_{k|k-1} H_k^T (H_k \tilde{P}_{k|k-1} H_k^T + R_k)^{-1}$
- $\hat{X}_{k|k} = \tilde{X}_{k|k-1} + K_k (Z_k - H_k \tilde{X}_{k|k-1})$
- $\hat{P}_{k|k} = \tilde{P}_{k|k-1} - K_k H_k \tilde{P}_{k|k-1}$

Step 6: Update X and P.

This process will get repeated until the end of the loop

3. Results and Discussions

All the algorithms mentioned above are tested for two videos (Dataset-I[19] and Dataset-II), Dataset I is less noisy and having occlusion problem. Dataset II is having a noisy complex background with the occluded target. The programming environment is MATLAB version 2013 b. The resolution of the video is 720 x 480 and the frame rate is 20 fps. The computer used is Intel core I5 with 4 GB RAM.

In Dataset I[19], a toy car is moving in the room and at certain frames, it is being occluded by the white box kept in the room. Dataset I is tested for all the algorithms and the value of parameters taken to implement MS-CBWH-AKFQR algorithm are, taking initial values of λ_1 (k) and λ_2 (k) as 0.8, value of ρ as 0.1, the value of ψ as 1000, B_T as 0.2, Δt as 0.05, initial value of dX and dY as 0.5.

It can be seen that for Figure 1, tracking window is tracking the target under normal condition but is missing the target under occlusion. It can be observed from the frames given in Figure 1 that the tracking window is tracking the car in the 386th frame and around 420th frame it is being occluded by the box and the window is unable to track the target in the forthcoming frame. In the 536th frame, the car is regaining the window, but after some time, it is losing the target around 961st frame due to occlusion.

In Mean Shift with CBWH and Kalman Filter (MSKF) type algorithms, from Figure 2 it can be seen that in the 368th frame the tracking window is tracking the car and around 420th frame it is being occluded by the box but due to Kalman filter, the car is again tracked efficiently in the 430th frame.

Figure 1. Image Frames of Dataset I for MS.

Figure 2. Image frames of Dataset I for MS-CBWH-K.

For all the six algorithms, Bhattacharya Coefficient is shown in Figure 3 and it can be observed that in occlusion, Bhattacharya Coefficient is getting decreased. For e.g. in the 450th frame of MS algorithm occlusion is there (Figure 1), Therefore in Figure 3 it can be seen that Bhattacharya Coefficient is low for MS and MS-CBWH in that region. For MSKF type algorithms, Bhattacharya Coefficient is increasing again after occluded frames because of Kalman filter shown in Figure 4(b).

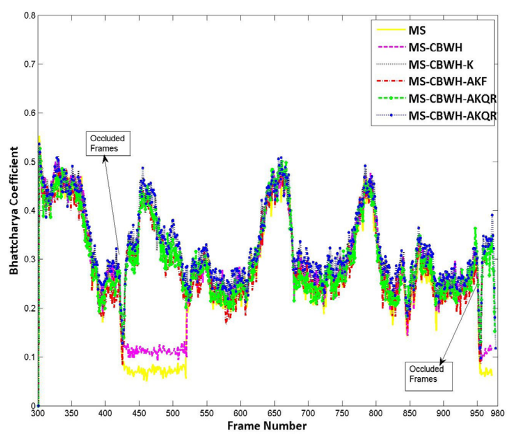

Figure 3. Bhattacharya Coefficient for Dataset I.

It can be observed from Figure 4(a) that, most of the time the Bhattacharya Coefficient for MS-CBWH-AKFQR algorithm is highest as compared to other algorithms, so it can be considered as the most suitable one for object tracking in the presence of occlusion. From Table 1, it can be observed that the mean value of Bhattacharya Coefficient for MSKF algorithms is more than simple MS or MS-CBWH algorithms. Amongst various MSKF algorithms, MS-CBWH-AKFQR has a highest mean value of Bhattacharya Coefficient than rest of the algorithms.

Table 1. Bhattacharya coefficient for Dataset I

Tracking Algorithms	Bhattacharya Coefficient
	Mean±Std.
MS	0.1827±0.1565
MS-CBWH	0.1892±0.1549
MS-CBWH-K	0.2137±0.1617
MS-CBWH-AKF	0.2034±0.1546
MS-CBWH-AKQR	0.2117±0.1573
MS-CBWH-AKFQR	0.2177±0.1476

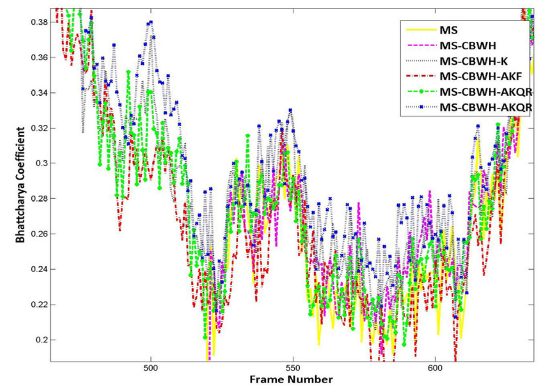

Figure 4. (a)Zoomed version of Figure 3.

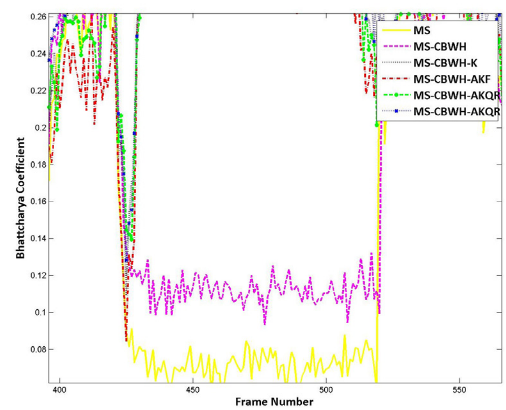

Figure 4. (**b**)Zoomed version of Figure 3.

In Figure 5, the time of execution for every algorithm is shown. It can be seen from Figure 6 that MS-CBWH-AKFQR is taking least time as compared to other MSKF algorithms. In Table 2, it can be observed that the execution time is comparatively less in MS-CBWH-AKFQR than other MSKF algorithms.

Table 2. Time taken for Dataset I

Tracking Algorithms	Execution time (sec)
	Mean±Std.
MS	0.002404±0.001968
MS-CBWH	0.002489±0.02481
MS-CBWH-K	0.003371±0.002588
MS-CBWH-AKF	0.003144±0.00249
MS-CBWH-AKQR	0.002875±0.002729
MS-CBWH-AKFQR	0.002761±0.002573

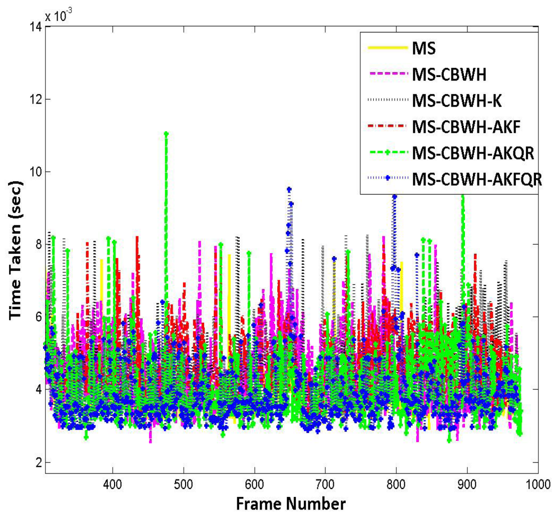

Figure 5. **Time Taken for Dataset I.**

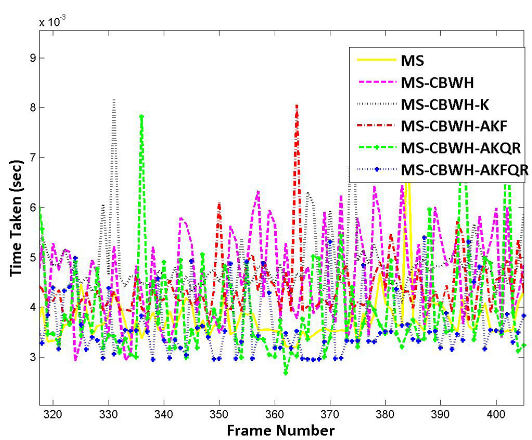

Figure 6. Zoomed Version of Figure 5.

In Dataset II, a boy is walking in CSIR-NAL, Bangalore garden having the complex background and having occlusion with trees. Dataset II is tested for all the Algorithms and values of parameters taken to implement MS-CBWH AKFQR are, taking initial values of λ_1 (k) and λ_2 (k) as 0.8, value of ψ as 400, value of ρ as 0.1, B_T as 0.4, dX and dY as 0.1 and Δt as 0.05, initial value of dX and dYas 0.1.

In the frames given in Figure 7, for Mean Shift (MS) and Mean Shift with CBWH (MS-CBWH) tracking, it is observed that in the 20th frame, the window is tracking the target but in the forthcoming frames target is being missed due to occlusion by trees.

In Dataset II, for Mean Shift with Kalman Filter (MSKF) type algorithms as shown in Figure 8, it can be seen that the boy is being occluded by trees in frame number around 250 but due to Kalman filter it is again getting tracked in the forthcoming frames unlike Mean Shift (MS) or CBWH (MS-CBWH) algorithm.

Due to partial occlusion with the trees, the tracking window is losing the target for MS and MS-CBWH algorithms. Hence after 120th frame, Bhattacharya coefficient for MS and MS-CBWH is less in Figure 9 and almost constant because of no movement of tracking window. In Figure 9, it can be seen that Bhattacharya Coefficient is low for MS and MS-CBWH for occluded frames but for MSKF algorithms, Bhattacharya Coefficient is increasing again after occluded frames because of Kalman filter shown in Figure 10(b). It can be inferred from Figure 10(a) and Table 3 that Mean Shift with CBWH and Kalman Filter with Adaptive F, Q, and R (MS-CBWH-AKFQR) has highest Bhattacharya coefficient for most of the frames. Hence, MS-CBWH-AKFQR is the most efficient algorithm. From Table 3, it can be observed that the mean value of Bhattacharya Coefficient with frame number is highest for MS-CBWH-AKFQR algorithm as compared to other algorithms.

Figure 7. Image frames of Dataset II for MS.

Figure 8. Image frames of Dataset II for MS-CBWH-K.

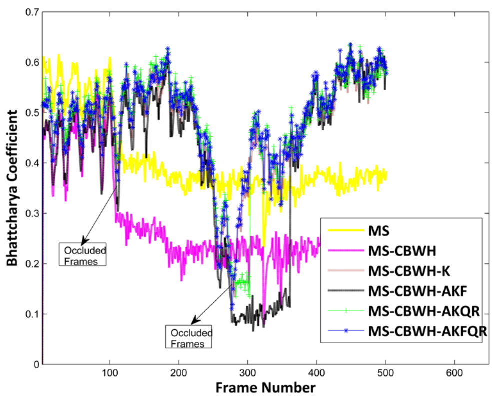

Figure 9. Bhattacharya Coefficient for Dataset II.

Table 3. Bhattacharya coefficient for Dataset II

Tracking Algorithms	Bhattacharya Coefficient Mean±Std.
MS	0.3716±0.07549
MS-CBWH	0.3996±0.08414
MS-CBWH-K	0.4729±0.1017
MS-CBWH-AKF	0.4095±0.1673
MS-CBWH-AKQR	0.4799±0.1138
MS-CBWH-AKFQR	0.4821±0.1021

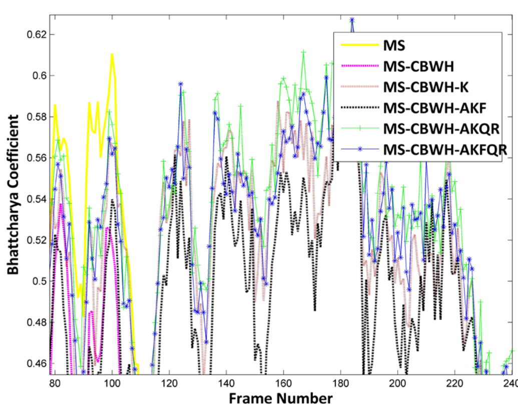

Figure 10. **(a)**Zoomed version of Figure 9.

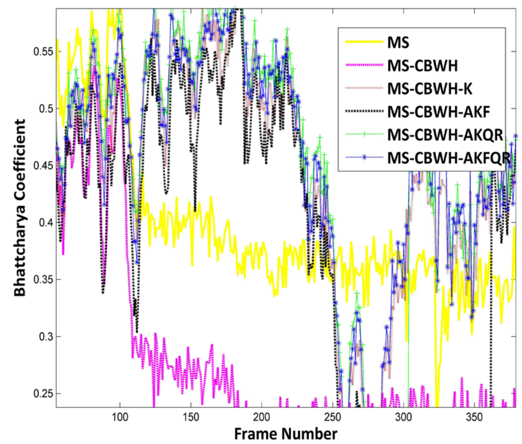

Figure 10. **(b)**Zoomed Version of Figure 9.

In Figure 11 the execution time for all the algorithms is shown. It can be seen from Figure 12 that MS-CBWH-AKFQR algorithm is taking least time as compared to other Mean Shift Kalman Filter (MSKF) algorithms. In Table 4, it can be observed that the mean time taken for all the frames is comparatively less in MS-CBWH-AKFQR than other MSKF algorithms and implementing the MS-CBWH-AKFQR is not taking much more time than the conventional Mean Shift (MS) algorithm.

Table 4. Time taken for Dataset II

Tracking Algorithms	Execution time (sec) Mean±Std.
MS	0.005181±0.004087
MS-CBWH	0.005286±0.001328
MS-CBWH-K	0.006550±0.001012
MS-CBWH-AKF	0.006342±0.001417
MS-CBWH-AKQR	0.005525±0.004218
MS-CBWH-AKFQR	0.005431±0.01049

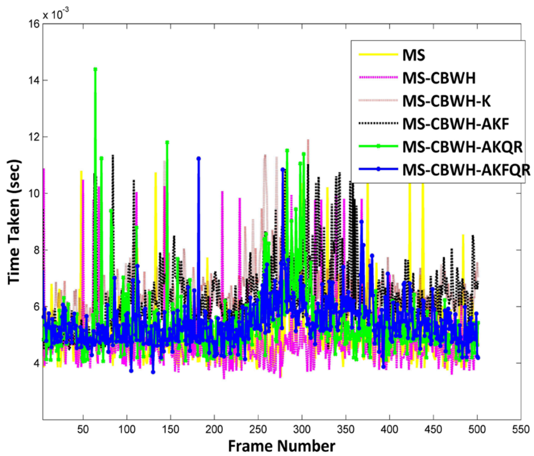

Figure 11. Time Taken for Dataset II.

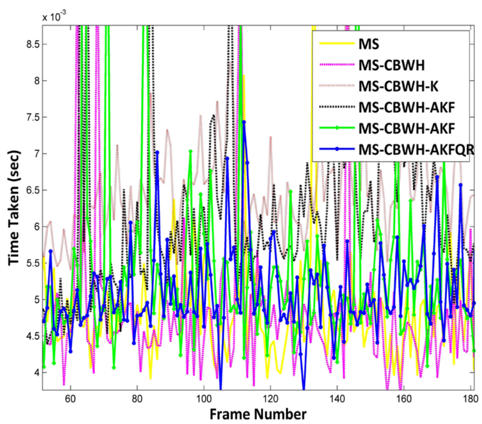

Figure 12. Zoomed Version of Figure 11.

From the above discussion, it can be inferred that Mean Shift with CBWH and Kalman with Adaptive F, Q, and R (MS-CBWH-AKFQR) is the better algorithm for tracking as the metrics like Bhattacharya Coefficient and Execution time are comparatively more and less respectively than other MSKF algorithms.

4. Conclusion

All algorithms are tested for two videos under the same environment and it is concluded that Mean Shift (MS) is successfully able to track the target under simple and less noisy environment but fails to track under complex and noisy environment and under the condition of occlusion. Mean Shift with CBWH (MS-CBWH) is able to track the target even in the complex and noisy environment but fails under occlusion. Simple Kalman Filter when implemented with Mean Shift and CBWH (MS-CBWH-K) is able to track the target even in occlusion but poses difficulty in varying background and also Q, R tuning of this filter is very difficult. Mean Shift with CBWH and Kalman with Adaptive QR (MS-CBWH-AKQR) is very efficient algorithm as it is able to tune itself according to Bhattacharya Coefficient automatically and the hectic problem of tuning is considerably less for this algorithm but due to constant system matrix parameters this algorithm suffered from the disadvantage of abrupt motion or velocity change problem. Mean Shift with CBWH and Adaptive F (MS-CBWH-AKF) has well- tuned and adaptive system matrix but its QR parameters are constant which makes it very difficult to tune and implement and it cannot adapt itself to changing background. Mean Shift with CBWH and Kalman with Adaptive F, Q, and R (MS-CBWH-AKFQR) is the better algorithm since it is taking the advantage of both the algorithms giving fairly better results than other algorithms.

Future work can be done in making the adaptive Kalman Filter as extended or unscented Kalman filter or the algorithm can be modified for tracking of multiple objects.

5. References

1. Indhu B, Naidu VPS. Gimbaled camera control for on-point target tracking. American Research Journal of Electronics and Communication Engg, C; 2015 Feb; 1(1):1–9.
2. Naidu VPS. Fusion of radar and IRST sensor measurements for 3D target tracking using extended kalman filter. Defense Science Journal. 2009 Mar; 59(2):175-82.
3. Salhi A, Jammouss AY. Object tracking system using cam-shift, mean shift, and Kalman filter. World Acad Sci Eng Technol. 2012 Apr; 6(4):674–9.
4. Jatoth RK, Shubhra S, Ali Ejaz. Performance comparison of Kalman Filter and mean shift algorithm for object tracking. I J Information Engineering and Electronic Business. 2013 Mar; 5(5):17–24.
5. Comaniciu D, Ramesh V, Meer P. Kernel-based objects tracking. IEEE: Trans Pattern Anal Mach Intel. 2003 Apr; 25(5):564–77.
6. Comaniciu D, Meer P. Real-time tracking of non-rigid objects using Mean Shift. IEEE Trans Pattern Anal Machine Intel. 2000; 24(5):603-19.
7. Comaniciu D, Ramesh V, Meer P. Real-time tracking of non-rigid objects using mean shift. IEEE Proceedings of Computer Vision and Pattern Recognition. 2000 Jun. p. 142–9.

8. Jeyakar J, Babu R, Ramakrishnan KR. Robust object tracking with the background-weighted local kernels. Computer Vision Image Underst. 2008 Dec; 112(3):296–309.
9. Ning J, Zhang L, Zhang D, Wu C. Robust mean-shift tracking with corrected background-weighted histogram. IET Computer Vision. 2012 Jan; 6(1):62–9.
10. Mazinan AH, Amir-Latifi A. Applying mean shift, motion information and Kalman Filtering approaches to object tracking. ISA Transactions. 2012 May; 51(3):485–97.
11. Zhao J, Qiao W, Men G. An approach based on mean shift and Kalman Filter for target tracking under occlusion. Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding; 2009 Jul 4. p. 12–5.
12. Object tracking using a Kalman filter. 2015. Available from: http://blog.cordiner.net/2011/05/03/object-tracking-using-a-kalman-filter-matlab/
13. Ahmed E, Ahmed A, Hadhoud M. A robust framework for object tracking based on corrected background weighted histogram mean shift and unscented Kalman Filter. IEEE IPAS'14: International Image Processing Applications and Systems Conference; 2014 Nov. p. 1–6.
14. Yang Y, Jia Y, Rong C. Object tracking based on corrected background-weighted histogram mean shift and Kalman Filter. Proceedings of the 2nd International Conference On Systems Engineering and Modeling; 2013 Apr. p. 687–2.
15. Karavasilis V, Nikou C, Likas A. Visual tracking by adaptive Kalman filtering and mean shift. International Journal of Computer and Electrical Engineering. 2010 May. p. 153–62.
16. Li X, Zhang T, Shen X. Object tracking using an adaptive Kalman Filter combined with mean. Shift Eng. 2010 Feb; 49(2):1-5.
17. Rene S, Georgakis C. How NOT to make the extended Kalman Filter fail. Ind Eng Chem Res. 2013 Jan; 52(9):3354–62.
18. Tian-Jian L, Zu-Tao Z. Adaptive double Kalman Filter and mean shift for robust fast object tracking. IJACT. 2013 Mar; 5(6):349–56.
19. Object tracking using a Kalman filter. 2015. Available from: http://cvlab.epfl.ch/teaching/topics/