

A Low Cost and Computationally Efficient Approach for Occlusion Handling in Video Surveillance Systems

Rakesh Chandra Joshi, Adithya Gaurav Singh, Mayank Joshi, Sanjay Mathur *

Deptt. of E&CE, Govind Ballabh Pant University of Agriculture and Technology, Pantnagar (India)

Received 19 September 2018 | Accepted 27 December 2018 | Published 11 January 2019



ABSTRACT

In the development of intelligent video surveillance systems for tracking a vehicle, occlusions are one of the major challenges. It becomes difficult to retain features during occlusion especially in case of complete occlusion. In this paper, a target vehicle tracking algorithm for Smart Video Surveillance (SVS) is proposed to track an unidentified target vehicle even in case of occlusions. This paper proposes a computationally efficient approach for handling occlusions named as Kalman Filter Assisted Occlusion Handling (KFAOH) technique. The algorithm works through two periods namely tracking period when no occlusion is seen and detection period when occlusion occurs, thus depicting its hybrid nature. Kanade-Lucas-Tomasi (KLT) feature tracker governs the operation of algorithm during the tracking period, whereas, a Cascaded Object Detector (COD) of weak classifiers, specially trained on a large database of cars governs the operation during detection period or occlusion with the assistance of Kalman Filter (KF). The algorithm's tracking efficiency has been tested on six different tracking scenarios with increasing complexity in real-time. Performance evaluation under different noise variances and illumination levels shows that the tracking algorithm has good robustness against high noise and low illumination. All tests have been conducted on the MATLAB platform. The validity and practicality of the algorithm are also verified by success plots and precision plots for the test cases.

KEYWORDS

Cascade Object Detector, Kalman Filter, Machine Learning, Occlusion Handling, Video Signal Processing, Video Surveillance.

DOI: 10.9781/ijimai.2019.01.001

I. INTRODUCTION

In recent times quest for development of intelligent security systems has become the need of the hour for making residential and office premises safer. Due to increasing threat from different types of activities leading to breach of security, it has become impossible for conventional security systems to detect such activities and alert security system in advance.

Thus, an increasing reliance on surveillance systems has resulted in need for better target detection and tracking techniques. Methods such as Radio Frequency Identification (RFID) tracking are not useful in preventing above mentioned situations; hence there is a need of wide area surveillance. Target tracking via image processing for a video surveillance system provides an attractive solution, which can efficiently track a specific target, record its position throughout the video stream and also analyze its motion pattern.

Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are; human-computer interaction [1], security and surveillance [2] [3], video communication and compression [4], augmented reality [5], traffic control [6] [7], medical imaging [8] [9],

video editing [10] [11], multimedia contexts [12] [13], complex object movements [14], video streaming [15], healthcare systems and smart indoor security systems. Video tracking is a time consuming process due to the amount of data that is captured and needs to be processed.

Further, the algorithm complexity increases if object recognition for tracking is also involved. The objective of video tracking is to maintain detectability of target object in consecutive video frames. To perform video tracking, an algorithm analyses sequential video frames and outputs the movement of targets between the frames. There are a variety of algorithms, each having its strengths and weaknesses. Considering the intended use, it is important to choose the algorithm best suited for the serving the purpose. Traditional tracking algorithms involve foreground extraction of the moving target from a static background and then tracks the coherent blobs of the target. Though, these algorithms are computationally efficient but track all the vehicles which exhibit motion in the stream. Similarly, other tracking algorithms like optical flow techniques as discussed in [16] and wavelet based vehicle tracking as illustrated in [17] also track all the vehicles which exhibit motion or all that are similar in appearance.

The lighting condition varies throughout the whole day (depending upon the weather condition for outdoor and lights for indoor). Low light conditions result into poor discrimination of objects from their background and sometimes lighting condition causes shadows or white-out effect. Most of the background subtraction techniques are sensitive to illumination change and it is difficult to handle the shade and shadow caused by the illumination change. Most algorithms which are able to handle these situations, need time on the order of several frames to estimate and train the background model [18]. Less work

* Corresponding author.

E-mail addresses: rakeshchandraindia@gmail.com (R. C. Joshi), agaurav51@gmail.com (A. G. Singh), mayankjoshi63@gmail.com (M. Joshi), mathur_sanjay@rediffmail.com (S. Mathur).

has been done in direction of tracking a specific vehicle throughout the video stream even in presence of other moving and similar in appearance vehicles, which is essentially the need for tracking an unidentified target.

The algorithm proposed in this paper has been developed keeping in mind the problem of tracking an unidentified vehicle entering a secured premise, such as a college campus. A basic complexity present in such situation is possible presence of other vehicles. The requirement thus becomes to track only the vehicle classified as unidentified and not the other vehicles irrespective of the fact that they are in motion or not. It is also required to track the target vehicle even through any possible occlusion and detect it post occlusion. The above mentioned problem can be divided into three different image processing procedures: Classification of vehicle if it is unidentified or not, target vehicle detection, target vehicle tracking, occlusion handling (if any). Vehicle can be classified as unidentified by a license plate reader as discussed in [19] which has been an established and widely used technology. The proposed algorithm, offers the approach how to detect, track and handle occlusion irrespective of presence of other vehicles in the video-stream. The problem at hand has been solved by adopting Kanade-Lucas-Tomasi (KLT) feature tracker with Cascaded Object Detector (COD) assisted by Kalman Filter (KF) which switches over the control of tracking according to the situations encountered. KLT tracker controls the tracking during no occlusion situations whereas COD takes over the flow when the target vehicle is under occlusion assisted by KF. This approach thus facilitates the realization of a smart vehicle tracking system and handles occlusion also. The algorithm has been tested successfully for tracking a single vehicle in 6 different cases of increasing complexity to test detection, tracking and occlusion handling capability of algorithm. Noise and Illumination variations for all cases have also been considered to test overall robustness of the algorithm.

The remainder of this paper is organized as follows. Section II describes related work in the area of the object tracking. Section III describes different approaches in vehicle tracking. Section IV introduces our proposed system. Experimental results are discussed in Section V. Finally, Section VI offers our conclusions.

II. RELATED WORK

In [20], Borisova *et al.*, proposed target tracking based method on object's shape when target has low contrast when compared to background. Maresca *et al.*, in [21] proposed the Matrioska tracking framework using Oriented FAST and Rotated BRIEF (ORB) features thus minimizing computational cost over Speeded Up Robust Features (SURF) features. Vehicle tracking using Fractional Feedback Kalman Filter was proposed by Kaur *et al.*, in [22] to improve the Kalman gain over traditional Kalman Filter. Feature tracking has been discussed by Ali *et al.*, in [23], to achieve multi object tracking of humans. In [24], Baheti *et al.*, discussed an automatic object tracking by a combination of Scale Invariant Feature Transform (SIFT) and Random Sample Consensus (RANSAC) which makes tracking invariant to translation and geometric transformations. Xia *et al.*, in [25] reliably track a target object in far field using SIFT features and RANSAC.

The real-time tracking with high accuracy in surveillance system is a challenging task. Most of the methods such as SANet [22], HCF [26], MDNet [27], SRDCF [28], MEEM [29], SINT [30], though give high distance precision but due to the computational load, these cannot be utilized in real-time tracking scenarios. In [31], a tracking method is proposed based on the object matching in every frame. It gives highly accurate results but comes with limitations on tracking speed which is due to the computational load in deep feature extraction requirements in each frame. On the contrary, Kernelized Correlation Filter (KCF) [32] tracks fast but it is unable to give much accurate results in short

period of time [33]. Similarly, the tracker in [34] requires many frames (normally 6) for initialization and tracks at low speed of 2 fps [35].

Fan and Ling [33] presented a parallel tracking and verifying framework, which consists of two major components, tracker and verifier. Tracker tracks in real time and verifier which runs at specific frame interval instead of each frame, check the tracking results and correct tracker if it is needed. Tracker adjusts the results according to the feedback provided by the verifier. But this method is highly dependent on value of frame interval for verification. If it is small computational load increases otherwise if it is large, the chances to deal the case of occlusion reduces. But our proposed algorithm, take necessary steps at the instant when object goes through occlusion. Kalal *et al.* [35] divided the tracking problem into tracking, learning and detection. Tracker provides the training data from learning component which estimates detection error, is used for updating the detector and the detector initiates again when tracker fails.

Correlation filter requires the information of object as well as the background or negative training data. In [36], to start a tracking process and to gather more information, an image patch is first cropped from the first frame as a sample patch got from initial object position and size information. There patch size is set to 2.5 times of the object size to provide context information about the foreground and background.

Li *et al.* [37] proposed a correlation filter tracker which also considers background hard-negative patches. Fully adaptive cluster method- Affinity Propagation (AP), is applied as a Background patch selection strategy. For AP method, clustering is executed with the help of real-value similarity matrix. In iteration process, cluster numbers and cluster centers are given. It uses one layer of Convolutional Neural Network (CNN). Although the accuracy is high in terms of success and precision rate, the computational load in training process is very high.

A tracking system has been discussed in [38] where a mixture of particle filter, and background modeling to track a group of people is used. Background modeling is helpful when application requires multiple targets to be tracked. An occlusion handling problem has been discussed in [39] using Gaussian background estimation but it does not track a single vehicle under occlusion. Single vehicle tracking under occlusion was also attempted to be solved in [40] where single targeted tracking has been achieved using color segmentation for segmentation of a particular vehicle and locking mechanism. Color segmentation however poses for high possibility of false matches as there can be multiple targets of same color.

Object tracking and data association comprise a set of computer vision techniques which deal with generation of the path and its trajectory in image plane by finding and locating the position of the object in every frame of the video sequence. A considerable research work has been done in both transportation and non-transportation applications. Different vehicle tracking approaches can be classified as follows.

A. Model Based Tracking

Three-dimensional model-based vehicle tracking algorithms have previously been discussed extensively in [41] and [42]. The techniques rely on recovering trajectories and models with high accuracy for a small number of vehicles. A major disadvantage of these approaches is the reliance on mathematically detailed geometric object models. It is not feasible to expect to have detailed models for all vehicles that could be found on the roadway as they can be so diverse in their geometry. A more practical approach will be to use template matching which has been discussed later as it relies on correlation and relative scores.

B. Region Based Tracking

In this approach, tracking system identifies a coherently moving connected region in the image, a 'blob', associated with each vehicle

and then tracks it over time [43]. Typically, the process is facilitated by the popular background subtraction technique. Foreground vehicles are detected by subtracting the input image frame from the current background estimate, looking for pixels where difference image is above some threshold and then finding connected components. This technique to track targets is widely used due to its low complexity.

However, this technique is more suited for the applications where all the moving targets are to be tracked, for example in studying flow of traffic. The problem discussed in this paper requires a single vehicle to be tracked even if other vehicles exhibit motion in the frame.

C. Active Contour Based Tracking

This tracking technique is based on active contour models, also known as snakes. This approach is similar to region based tracking. The underlying principle is to retrieve a bounding contour of the object and update it dynamically [44] [45]. According to the contour position in previous and current frame the object is tracked. This technique requires some overlapping of the object region in between previous frame and the current frame. This technique, however, generates significant measurement errors when the target undergoes an occlusion. Also, the problem of tracking the targeted vehicle due to the presence of other vehicles still remains unresolved since active contour method extract fine details about the boundary of the objects with respect to all the background disturbances.

D. Kernel Based Tracking

Kernel by definition means “the central part of something”. When it comes to object tracking, kernel is the central component of the target being tracked. A kernel based tracking algorithm is discussed in [46] in which an isotropic kernel is marked with the target spatially to generate the similarity function and according to that object is localized in an image. Recognition and classification based methods such as template matching, cascade object detection and support vector machine are among the kernel based tracking methods. All of these methods provide the center point of the rigid body which is to be tracked.

These methods provide an attractive option for tracking a single vehicle as they return a single best match; however, this best match is heavily based on the appearance of the target. Since, vehicles like cars have similar appearance, the possibility of false match increases. Also, applying search on every frame over all the pixels increases the computational cost unless search for the target can somehow be made local.

E. Feature Based Tracking

A different approach for tracking an object is feature based tracking which involves finding distinguishable lines, points and corners, considered as features of the target object in the current frame and then matching these features in the next frame. This matching allows the computation of motion model of the vehicle which can be used for tracking the vehicle [47][48]. This method has many advantages. It stabilizes the computational cost of the system. In the presence of partial occlusion some features of the moving vehicle remain visible. Also, feature tracking is possible in low illumination.

From the review of related work, it is found that it is difficult to achieve high precision in real-time tracking applications. Moreover, post occlusion detection accuracy is further affected in such situations. From the forgoing discussion we also concluded that a simultaneous operation of different algorithms is required to track a single targeted vehicle. On comparing all the approaches of tracking approaches feature based tracking is a promising choice as it stabilizes computational cost, does not require foreground extraction and doesn't require detailed geometric models.

III. PROPOSED SYSTEM

The paper presents an approach to track an already classified unidentified vehicle and to handle its occlusion. Our paper proposes utilization of IP camera using a local dedicated network having wireless routers and mobile hotspots to capture the scene unlike in [42] where frame capturing is done via IP camera using an internet connection which makes the system speed and network connectivity dependent.

We propose to detect the target using spatially local template matching technique as discussed later which returns only one detected target whose features are then tracked without any locking mechanism thus decreasing the computational cost of the system. Another tracking technique has been discussed in [49] using KLT feature tracker over SURF features. Though SURF features are reliable, single targeted tracking requires maximum features to be extracted. As discussed in [50], a much higher number of features are extracted using Minimum Eigen Value of interest points used in our algorithm as compared to the number of features extracted using SURF algorithm.

In Fig. 1, the flow chart elaborates the algorithm operation, comprehensively.

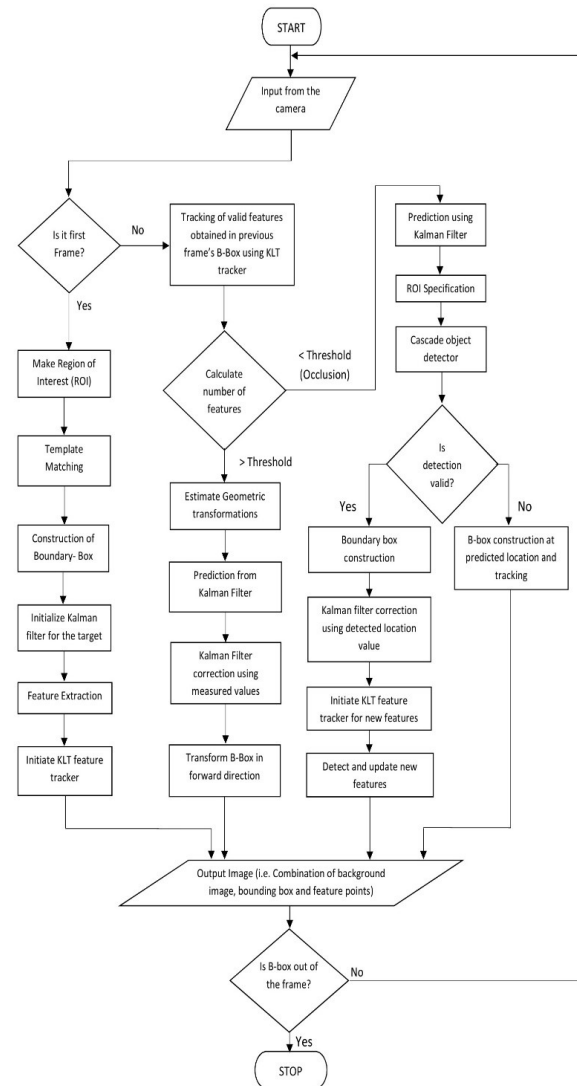


Fig. 1. Flow Chart of Proposed Method.

In the last condition of flow chart, STOP signifies that the bounding box around the targeted vehicle goes out of the frame. Then the system is reset and all the previous values are deleted.

Unidentified vehicle tracking problem is divided in terms of different situations that are required to be handled.

A. Detection In First Frame

In case of vehicle classified as unidentified, camera suitably positioned takes over its tracking operation. In order to be tracked it is essential to identify the location of this vehicle in the first frame correctly. The detection is done using template matching method by dividing the frame into a number of square windows and obtaining some score for each window such as suggested in [51]. However, searching the whole frame for the template can result in false detection due to possible presence of other vehicles. This issue can be resolved by searching the template locally instead of searching it globally. Thus template search is applied only in the region of pixels where geographical location of the entrance is correlated in the frame, as it can be safely assumed that the target car in the first frame will be in vicinity of the entrance. The template matching algorithm used in our study is based on the sum of absolute differences to find the best match given by (1).

$$d(A, B) = \sum_i \sum_j |A(i, j) - B(i, j)| \quad (1)$$

where A = Input frame, and B = Template and i, j = coordinate of pixel.

The tracking bounding box is constructed around the kernel for the window where $d(A, B)_{\min}$ is recorded.

B. Tracking in Subsequent Frames

Once the position in first frame is detected successfully, bounding box is constructed on the detected location to indicate the car in output frame. Feature tracking has been recognized as very reliable method to track targets in computer vision. Feature tracking relies on the concept of finding distinguishable points, corners & lines, called features, of the target. These features are then matched in every subsequent frame and a geometric transformation motion model is estimated which facilitates tracking of the target.

The proposed algorithm also detects features inside the bounding box, i.e. on the car, by finding distinguished points and corners. As bounding box is constructed with the help of template matching approach, so it works better than those methods such as TLD [35], LOT [52], DFT [53], CXT [54] which has reliability on exact boundary around the target in first frame or fixed-size patches. These methods degrade their performance on increasing scaling of the patch at initialization and are more sensitive to the background clutter [55]. Features can be detected by various methods as in [56] [57] [58] [59]. Good features are located by examining the minimum Eigen value of each 2×2 gradient matrix of every pixel in region of interest as proposed in [60]. This set of feature points is then tracked by the Pyramidal Implementation of the KLT feature tracking algorithm [61] in each subsequent frame which minimizes the residual function e , throughout the down-sampled image pyramids, defined in (2) as follows

$$\begin{aligned} e(d) &= e(d_x, d_y) \\ &= \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (I(x, y) - J(x + d_x, y + d_y))^2 \end{aligned} \quad (2)$$

where, $u = [u_x, u_y]^T$ is a point on image I , whose location, $v = u + d = [u_x + d_x, u_y + d_y]^T$ in subsequent frame J is to be found out.

This algorithm is often used for short-term tracking as a part of larger tracking framework. Number of image pyramids formed by down-sampling previous frame levels is chosen by default to be three. This algorithm also has the capability of marking each feature as either valid or invalid for each frame. Once the feature tracker maps the features from one frame to next, geometric transformation is computed using the similarity projection as given by (3) [62].

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a \cos \theta & -a \sin \theta & t_x \\ a \sin \theta & a \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3)$$

Here, $[x \ y]$ are the coordinates of the feature tracked in first frame, $[x' \ y']$ are the coordinates of the same feature in the next frame, t_x and t_y are the translation in x and y axis respectively, θ is the projection angle and a is the scaling factor. For similarity geometric transformation, at least 2 pairs of matched features are required to compute all 4 degree of freedom. Four degree of freedom meant for translation (t_x, t_y), scaling (a) and rotation (θ). For affine transformation two more degree of freedom added i.e. aspect ratio and shear. On the basis of transformation model thus computed the tracking marker/bounding box is also transformed forward. Only the valid feature points are considered to estimate the geometric transform of the target.

In the first frame the size of the bounding box is directly correlated to the size of template which in turn is dependent on the geographical position where the camera is installed i.e., the size of template in first frame will be decided on case to case basis during the installation of the system, while in subsequent frames by finding the geometric transform between the feature points using the similarity projection (which encompasses scaling transform) the size of bounding box is adjusted, hence even if scale of vehicle size changes during tracking bounding box size is also scaled. As the tracking progresses over time, points can be lost due to occlusions. If a condition arises where number of valid points fall below a threshold, they need to be reacquired to track the object further. This is the condition of occlusion and needs a different technique to be handled.

C. Occlusion Handling

In computer vision, occlusion is the condition when the target being tracked is hidden by another object in the frame. The problem becomes more complicated when the target disappears and reappears after a brief occlusion and has to be tracked again. Partial occlusion (target size > occluding object's size) and complete occlusion (target size < occluding object's size) cases are shown in Fig. 2.



Fig. 2. Vehicle under (a) partial occlusion (b) complete occlusion.

In feature tracking based algorithm like ours, occlusion is characterized by an extensive loss of features as the target hides. To solve the problem of occlusion it is important to redetect and specify new features if the target re-appears again. Redetection in our algorithm is in the form of a degenerate decision tree, as shown in Fig. 3, of classifiers cascaded together and applied to every sub-window of the frame to classify them as a positive or a negative result as proposed in [63].

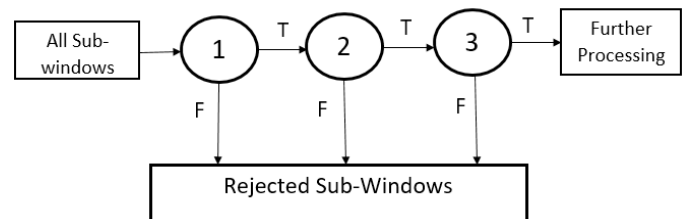


Fig. 3. Schematic representation of the Degenerate Decision Tree based Cascade Object Detector as proposed in [63].

The classifiers are trained with a dataset of cars named GRAZ-02 as developed in [64]. However, just like the first frame detection, the probability of a false match due to the presence of other cars when applied to the frame globally is high. Again, to eliminate the possibility of a false match it is important to search the target after occlusion locally, only at the expected location of target in the next frame, instead of searching it globally in the whole frame.

However, to predict this expected location in next frame it is necessary to compute a motion model of the target according to the history of its motion. The position of the car is a time dependent state vector. Kalman filter [65] is an efficient recursive computational solution to track a time dependent state vector, like position vector of a car, with equations of motion using least-squares method. Kalman filter assists and computes a state transition model to predict the dynamic position of the tracked target defined by following equations.

$$x_k = Ax_{k-1} + w_{k-1} \quad (4)$$

$$z_k = Hx_k + v_k \quad (5)$$

Here, x_k = predicted position of target using previous state; A = State Transition Matrix; x_{k-1} = previous position of target; w_{k-1} = process noise; H = Measurement Matrix; v_k = Measurement Noise; z_k = Measurement.

For each frame x_k is estimated depending on x_{k-1} . The model is then corrected using the measurement z_k . This process is then performed recursively to enhance the model's accuracy.

Condition of occlusion is characterized by the absence of z_k . In this condition, x_k is computed for next frame and the classifier searches for positive result only in the enlarged region of interest in the local neighborhood of x_k which makes our algorithm capable of tracking and detecting the object which is moving with slow or fast speed w.r.t. predicted speed and can also handle the drift or movement in upper and lower coordinated points as shown in Fig. 4. The region of interest (ROI) is greater than the maximum size of any vehicle that may be viewed by the camera and it is just enough in size so that it recognizes the vehicle coming out of the occlusion. ROI constructed in our proposed algorithm is 2.5 times greater than the bounding box.

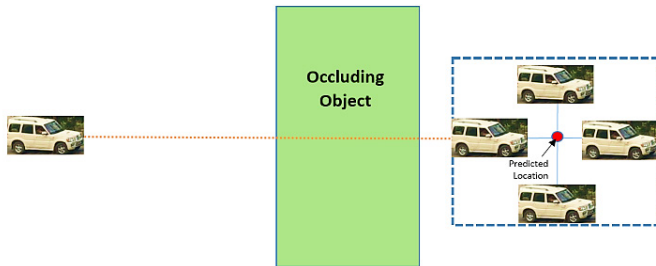


Fig. 4. Enlarged ROI at the time of occlusion.

If a positive result is retrieved new features are detected in the returned bounding box and tracking continues. As positive result will be achieved for the vehicle, most of the features are of vehicle. But in some critical cases, it can be possible that at the end of occlusion there is the inclusion of features of the occluding objects inside the bounding box, but those features are eliminated as *correction* is done in terms of KLT feature updating and, at the end, it will track only the object features and static object or occluding object's features get lost. If a positive result is not retrieved, prediction for next frame is computed and classifier searches again in the new region of interest. The process is recursive until vehicle is detected again after occlusion. If in case of occlusion, COD is not applied, then it can track accurately only those vehicles which are exact at the point of prediction x_k . But such case is not always possible and extended ROI concept will also

not be applicable. One solution is to make buffer storage of cropped image at certain interval, around the bounding box, and match that patch at the instant when occlusion occurs around the predicted point or on the whole frame. But additionally, computational overload will also be increased.

Vehicle can also be tracked as usual if there are object's features, even in those cases where it exhibits the non-linear motion or when linear motion of vehicle in ground plane is projected non-linear in image plane due to the distance between vehicle and image plane.

In Fig. 5, working of KLT tracker, KF and COD is shown in 2 different cases, i.e. with and without occlusion.

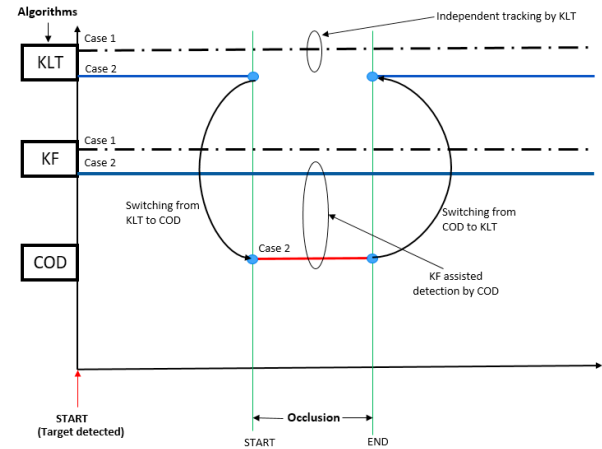


Fig. 5. State Diagram of KLT, KF, COD working together in 2 different cases, i.e. Case 1: without occlusion, Case 2: occlusion (partial and complete).

The next objective is to track the vehicle under different scenarios.

IV. RESULTS

The proposed algorithm has been tested on six different cases of increasing complexity. We have used an Android mobile phone camera (13 megapixels) with IP cam application for capturing the video which is connected to a server cum processing system operating on a 1.5 GHz AMD processor, with 4GB RAM and 64 bit operating system, via Mobile Hotspot. Image processing is done on MATLAB 2016 version. Frame size of the video is 1280×720 and more.

Parametric studies of different Gaussian noise variance and different illumination level have also been performed for the no occlusion, partial occlusion and complete occlusion cases (in all cases where other vehicles are also present). In all the cases variance of noise and illumination level have been varied, the error between the coordinates of tracked vehicle in the case under study and the coordinates obtained for the reference conditions, i.e. zero noise and normal day light illumination (approximately 111,000 lux); have been plotted for each frame. The variance has been kept in increasing trend from 0.01 to 0.125, whereas illumination with respect to the reference condition is varied synthetically by reducing it from 100% illumination to 10% illumination. The closer the recorded tracking error is to zero, the better is the tracking for that frame.

We have taken such cases where there are less number of frames (on an average 50 frames) before occlusion to make cases more complex to check robustness of algorithm. Fig. 6 shows the images with varying Gaussian noise variance 0.01(default value), 0.075 and 0.125 with respect to the original scene.

Although the bounding box size (width) will vary in some cases due to the distance of the camera and the vehicle, in our case it was observed that an average size of 150 pixels is the size of the bounding

box during tracking. In all the cases, the deviation (error) from ideal case was not more than 14% and therefore a tolerance level has been set at (\pm) 20 pixels. Even in the case when the bounding box size was as large as 300 pixels, the tracking error was found to be less than (\pm) 20 pixels which shows the high accuracy of our algorithm.

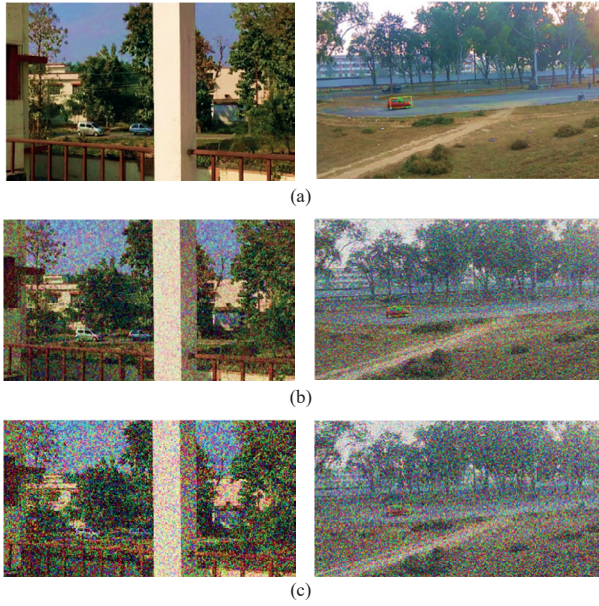


Fig. 6. Images at (a) default 0.01 Gaussian noise variance, (b) with 0.075 Gaussian noise variance, (c) with 0.125 Gaussian noise variance.

Fig. 7, 8, 10, 11, 13, 14 show tracking of the vehicle in six different cases as seen on the monitor with increasing complexity levels.

A. Single Vehicle Without Occlusion



Fig. 7. Tracking of the unidentified target vehicle.

B. Vehicle In Presence of other Vehicles Without Occlusion



Fig. 8. The tracked vehicle in presence of other vehicle.

In the case of Vehicle tracking without occlusion for which error plots are depicted in Fig. 9, system worked perfectly fine when Gaussian noise variance was less than 0.125. With the noise variance crossing 0.125 tracking error increased beyond 20 pixels. However, tracking of vehicle under different illumination levels was found largely illumination invariant under the no occlusion condition as seen in Fig. 9(b).

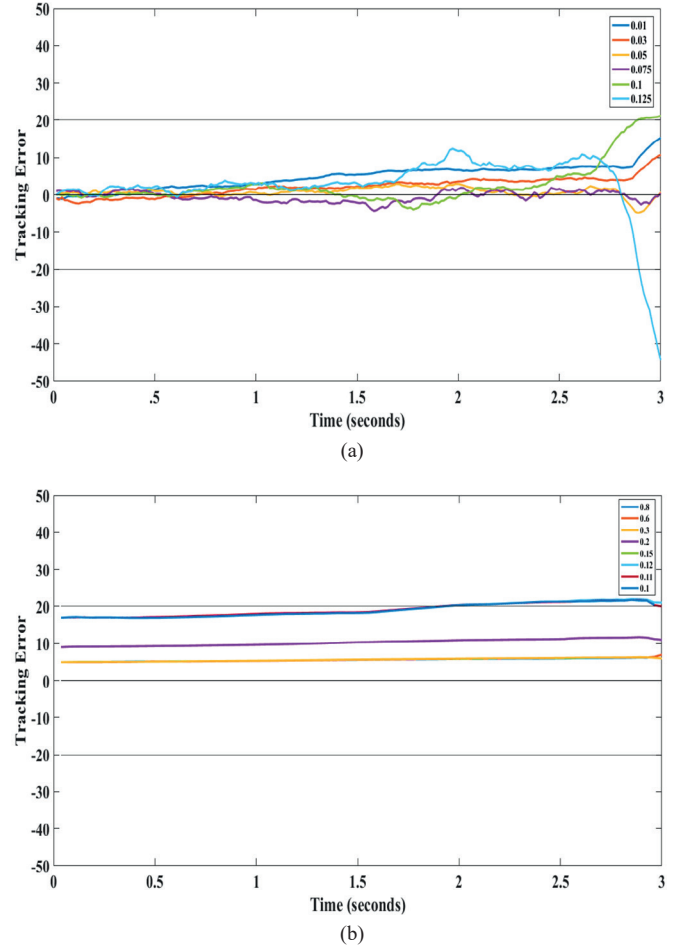


Fig. 9. (a) Tracking Error with different Gaussian Noise Variance (without occlusion) (b) Tracking Error with different Illumination Level (without occlusion).

C. Single Vehicle Undergoing Partial Occlusion



Fig. 10. Tracking of the vehicle before, during and after undergoing partial occlusion.

D. Vehicle in Presence of Another Vehicle Undergoing Partial Occlusion



Fig. 11. Tracking of the vehicle before, during and after undergoing partial occlusion in presence of other vehicle.

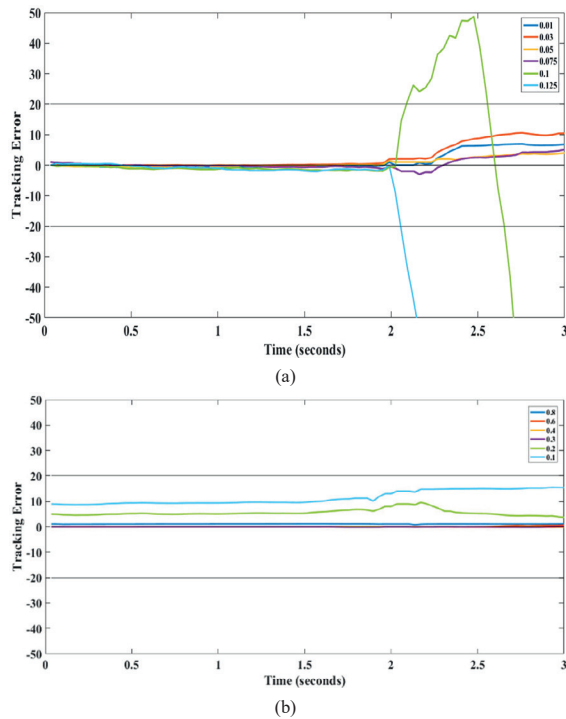


Fig. 12. (a) Tracking Error with different Gaussian Noise Variance (partial occlusion) (b) Tracking Error with different Illumination Level (partial occlusion).

In the case of Vehicle tracking with partial occlusion, system worked well between the ± 20 pixels limit until the noise variance was below 0.075 as illustrated in Fig. 12(a), at 0.1 variance; the system tracks the vehicle accurately before the occlusion but the tracking error increases beyond the ± 20 pixels limit after occlusion which indicates that algorithm is more susceptible to errors after occlusion due to classification by cascade classifiers. At 0.125 variance, the system fails to track after occlusion, which is an expected result. As in the previous case, system again proved to be illumination invariant, shown in Fig. 12(b) with a slight error after occlusion but still inside the ± 20 pixels limit even when illumination is as low as 10% or 0.1 times the reference illumination. The scene is very complex as it consists of a tree as an occluding object instead of a simple pole. In this scene features not

reached threshold instantly, as features not wiped out on a vertical line. Hence, command transfer to the COD is delayed by few frames. In this case, our expanded ROI approach is able to handle these type of cases.

E. Vehicle Alone Undergoing Complete Occlusion

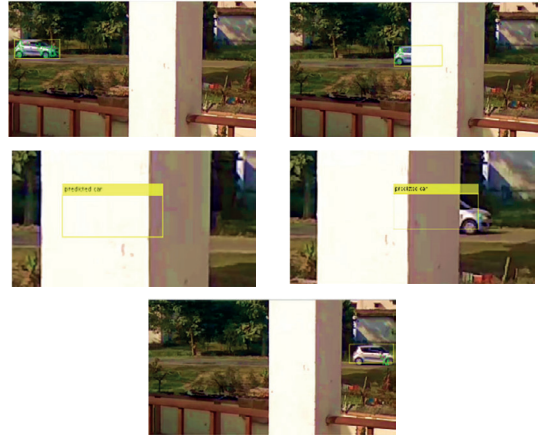


Fig. 13. Tracking of the vehicle before, during and after undergoing complete occlusion.

F. Vehicle in Presence of other Vehicle Undergoing Complete Occlusion

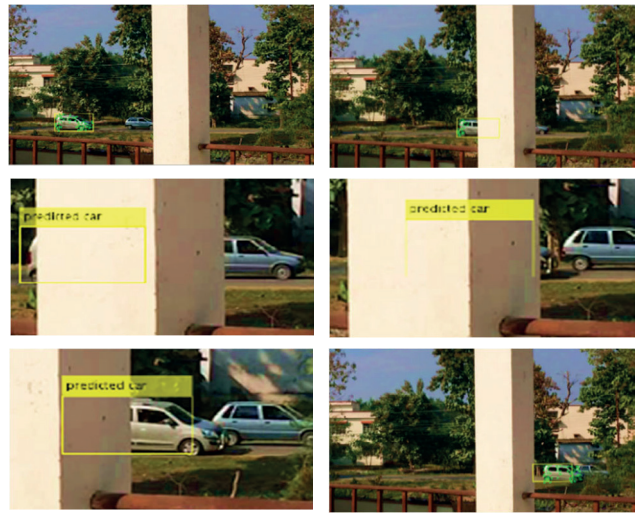
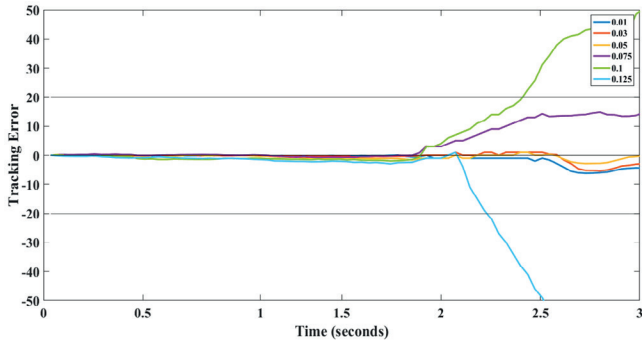


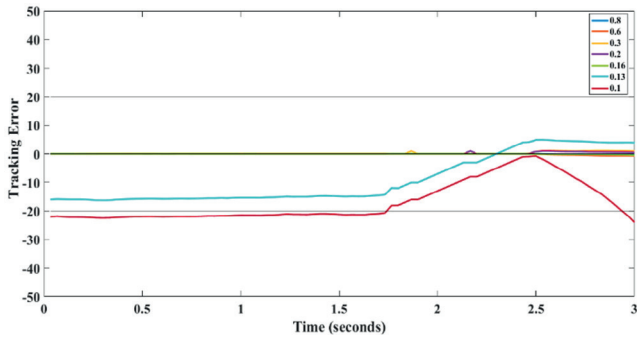
Fig. 14. Tracking of the vehicle before, during and after undergoing complete occlusion in presence of other vehicle.

In the case of Vehicle tracking with complete occlusion, our algorithm is capable to handle the occlusion even after short duration of learning and the system is robust to noise until variance value of 0.075. Although the tracking error increases after occlusion, it is in between the ± 20 pixels limit for variance value of 0.075. For variance equal to 0.1, the system loses tracking earlier and shows an erratic behavior after occlusion, i.e. obtained position of the target leads the reference position and then after some frames, obtained position starts lagging the reference position of the target, for 0.125, system fails after occlusion due to very high noise as depicted in Fig. 15(a). On varying the Illumination, the system worked fine with a slight tracking error after occlusion until 17% illumination, at 15% illumination level; there was a tracking error of approximately (-)18 pixels from the first frame, which was reduced by Kalman filter during complete occlusion and was reduced to (+)7 pixels after occlusion, as shown in Fig 15(b). At 10% illumination or 0.1 times reference illumination

level, tracking error was slightly higher than (-)20 pixels and during occlusion; Kalman filter reduced the tracking error to approximately (-)6 pixels. The tracking error recorded can be attributed to excessively low illumination level.



(a)



(b)

Fig 15. (a) Tracking Error with different Gaussian Noise Variance (complete occlusion) (b) Tracking Error with different Illumination Level (complete occlusion).

The tracking is judged basically on two parameters- Success and Precision rate. Precision plot represents the percentage of frames which are within the estimated ground truth threshold varying from 0 to 50. Precision score is calculated where difference of tracked object location and ground truth location is within the distance of 20 pixels. Success plot represents the percentage of frames where overlap score is greater than the threshold. Overlap score, S is calculated as-

$$S = \frac{|B_t \cap B_g|}{|B_t \cup B_g|} \quad (6)$$

where, B_t and B_g are the tracked and ground truth bounding box respectively, $|\cdot|$ represents the number of pixels in a region [55][66]. Success score is calculated for 0.5 overlap threshold. Precision plots and success plots are shown in Fig. 16 and 17 respectively.

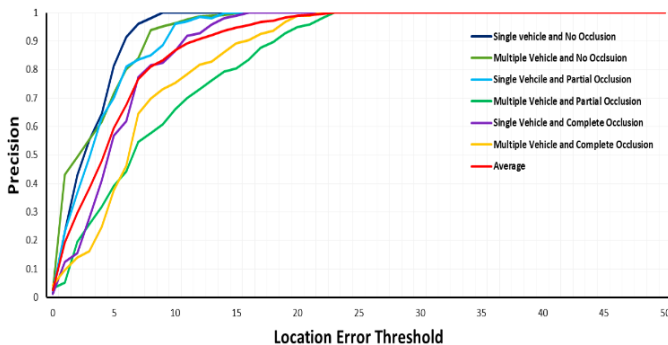


Fig. 16. Precision Plot.

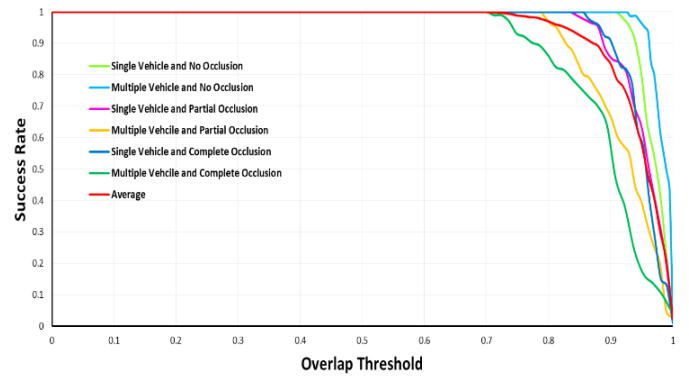


Fig. 17. Success Plot.

The precision score and success score are resulted as 0.9861 and 1.0 respectively. Hence, tracking is done very well with high accuracy.

Table I and Table II show the Maximum Tracking Error; the peak value of tracking error in different cases.

TABLE I. MAXIMUM TRACKING ERROR IN 3 DIFFERENT CASES WITH INCREASING NOISE VARIANCE

Gaussian noise variance	Maximum tracking error (pixels)		
	Without occlusion	Partial occlusion	Complete occlusion
0.01	12	6	(-)6
0.03	8	10	(-)5
0.05	(-)5	4	(-)3
0.075	(-)7	6	16
0.1	22	48	43
0.125	(-)25	Above (-)50	Above (-)50

TABLE II. MAXIMUM TRACKING ERROR IN 3 DIFFERENT CASES WITH DECREASING ILLUMINATION LEVEL

Illumination level (with respect to reference level)	Maximum tracking error (pixels)		
	Without occlusion	Partial occlusion	Complete occlusion
80%	5	3	(-)3
60%	6	2	(-)2
40%	Nil	2	Nil
30%	5	3	(-)4
20%	11	8	3
17%	Nil	Nil	3
15%	21	Nil	(-)4
13%	Nil	Nil	(-)16
12%	22	Nil	Nil
11%	21	Nil	Nil
10%	22	16	(-)23

From an overall analysis, it can be deduced that the proposed algorithm works best in the conditions where Gaussian noise variance is lower than or equal to 0.075 and illumination is greater than 10% or 0.1 times the reference illumination level.

V. CONCLUSION

To commercially develop a smart surveillance systems that can track an unidentified vehicle and can also detect if it is passing through some occlusion, it is important to design an algorithm which has the capability to track and analyze a single moving vehicle through a video stream even when other vehicles are also present in the stream. This paper proposed an algorithm which solves the above problem by a coherent operation of different tracking and recognition techniques such as Feature Tracking, Template Matching, Cascade Object Detection and Motion estimation using Kalman Filter. The algorithm was tested successfully for 6 cases of different complexity in real time. The performance was analyzed at high noise and low illumination and the algorithm was found to be exhibiting good performance even at high Gaussian noise with a variance of 0.075 and low illumination of 10%. It is able to handle occlusion in a short interval of learning. The precision and success score are 0.9861 and 1.0 respectively which signifies very good tracking.

Our proposed system is completely wireless and works on real time, even more frames can be handled if we reduce the frame size. It is easy to install and cost effective as it is not using other things such as FPGA boards or server based system. System is resistive to motion blur; capable of handling high frame rate (30 fps or more) depending on the connectivity between IP camera and router/hotspot. We have the flexibility to use it with internet or without internet i.e. If we want to use for the local surveillance we can use Wi-Fi network or mobile hotspot without the internet and if we require the surveillance from far place or any part of the world then we can use the network with internet. Surveillance system is secured as the network made through mobile hotspot/ Wi-Fi is password protected (WPA security).

VI. FUTURE SCOPE

A series of modifications have to be made in above algorithm to make it more efficient to be implemented in a surveillance system to solve other problems. These include solving the problem of inter vehicle occlusion, introducing PTZ (Pan-Tilt-Zoom) camera to make more efficient tracking so that less number of cameras will be required, communication with other cameras when the tracked object goes out of the field of view of one camera to other camera's field of view. Our developed algorithm can also be utilized to track human and other objects during occlusion.

ACKNOWLEDGMENT

This work was supported by the scholarship provided to the first author for postgraduate studies by Ministry of Human Resource Development (MHRD), Government of India.

REFERENCES

- [1] Bulling and H. Gellersen, "Toward Mobile Eye-Based Human-Computer Interaction," in *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 8-12, October-December 2010.
- [2] T. Ko, "A survey on behavior analysis in video surveillance for homeland security applications," 2008 37th IEEE Applied Imagery Pattern Recognition Workshop, Washington DC, 2008, pp. 1-8.
- [3] A. Ratte and V. Pankajakshan, "Tucker tensor decomposition-based tracking and Gaussian mixture model for anomaly localisation and detection in surveillance videos," in *IET Computer Vision*, vol. 12, no. 6, pp. 933-940, 9 2018.
- [4] E. Soyak, S. A. Tsaftaris and A. K. Katsaggelos, "Low-Complexity Tracking-Aware H.264 Video Compression for Transportation Surveillance," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1378-1389, Oct. 2011.
- [5] H. Kim, J. L. Gabbard, A. M. Anon and T. Misu, "Driver Behavior and Performance with Augmented Reality Pedestrian Collision Warning: An Outdoor User Study," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1515-1524, April 2018.
- [6] D. Beymer, P. McLauchlan, B. Coifman and J. Malik, "A real-time computer vision system for measuring traffic parameters," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, 1997, pp. 495-501.
- [7] Q. Zou, H. Ling, Y. Pang, Y. Huang and M. Tian, "Joint Headlight Pairing and Vehicle Tracking by Weighted Set Packing in Nighttime Traffic Videos," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1950-1961, June 2018.
- [8] A. Ghoneim, G. Muhammad, S. U. Amin and B. Gupta, "Medical Image Forgery Detection for Smart Healthcare," in *IEEE Communications Magazine*, vol. 56, no. 4, pp. 33-37, April 2018.
- [9] S. Harput et al., "Two-Stage Motion Correction for Super-Resolution Ultrasound Imaging in Human Lower Limb," in *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 65, no. 5, pp. 803-814, May 2018.
- [10] W. Paier, M. Kettern, A. Hilsmann and P. Eisert, "A Hybrid Approach for Facial Performance Analysis and Editing," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 784-797, April 2017.
- [11] Feng Wang, Chong-Wah Ngo and Ting-Chuen Pong, "Gesture tracking and recognition for lecture video editing," *Proceedings of the 17th International Conference on Pattern Recognition*, 2004. ICPR 2004, pp. 934-937 Vol.3.
- [12] A. Y. Yang, S. Iyengar, S. Sastry, R. Bajcsy, P. Kuryloski and R. Jafari, "Distributed segmentation and classification of human actions using a wearable motion sensor network," *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK, 2008, pp. 1-8.
- [13] A. Jalal and S. Kamal, "Real-time life logging via a depth silhouette-based human activity recognition system for smart home services," *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Seoul, 2014, pp. 74-80.
- [14] Y. Song, J. Tang, F. Liu and S. Yan, "Body Surface Context: A New Robust Feature for Action Recognition From Depth Videos," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 952-964, June 2014.
- [15] A. Jalal, S. Kamal and D. Kim, "Shape and Motion Features Approach for Activity Tracking and Recognition from Kinect Video Camera," *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, Gwangju, 2015, pp. 445-450.
- [16] T. Senst, V. Eiselein and T. Sikora, "Robust Local Optical Flow for Feature Tracking," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1377-1387, Sept. 2012.
- [17] J. B. Kim, C. W. Lee, K. M. Lee, T. S. Yun and H. J. Kim, "Waveletbased vehicle tracking for automatic traffic surveillance," *Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology. TENCON 2001 (Cat. No.01CH37239)*, 2001, pp. 313-316 vol.1.
- [18] W. Kim and C. Jung, "Illumination-Invariant Background Subtraction: Comparative Review, Models, and Prospects," in *IEEE Access*, vol. 5, pp. 8369-8384, 2017.
- [19] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen and P. Shi, "An Algorithm for License Plate Recognition Applied to Intelligent Transportation System," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 830-845, Sept. 2011.
- [20] V. Borisova and G. D. Yakovlev, "Target tracking based on object's shape," *2016 11th International Forum on Strategic Technology (IFOST)*, Novosibirsk, 2016, pp. 397-400.
- [21] M. E. Maresca and A. Petrosino, "The Matrioska Tracking Algorithm on LTDT2014 Dataset," *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, 2014, pp. 720-725.
- [22] H. Kaur and J. S. Sahambi, "Vehicle Tracking in Video Using Fractional Feedback Kalman Filter," in *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 550-561, Dec. 2016.
- [23] M.M. Naushad Ali, M. Abdullah-Al-Wadud, Seok-Lyong Lee, Multiple object tracking with partial occlusion handling using salient feature points, *Information Sciences*, Volume 278, 2014, Pages 448-465, ISSN 0020-0255, <https://doi.org/10.1016/j.ins.2014.03.064>.

- [24] Baheti, U. Baid and S. Talbar, "An approach to automatic object tracking system by combination of SIFT and RANSAC with mean shift and KLT," 2016 Conference on Advances in Signal Processing (CASP), Pune, 2016, pp. 254-259.
- [25] Y. Xia, G. Li, R. Chen and P. Yu, "A object tracking method in two cameras with common view field," 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, 2017, pp. 1-5.
- [26] H. Fan and H. Ling, "SANet: Structure-Aware Network for Visual Tracking," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, 2017, pp. 2217-2224.
- [27] C. Ma, J. Huang, X. Yang and M. Yang, "Hierarchical Convolutional Features for Visual Tracking," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 3074-3082.
- [28] H. Nam and B. Han, "Learning Multi-domain Convolutional Neural Networks for Visual Tracking," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 4293-4302.
- [29] M. Danelljan, G. Häger, F. S. Khan and M. Felsberg, "Learning Spatially Regularized Correlation Filters for Visual Tracking," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 4310-4318.
- [30] Zhang J., Ma S., Sclaroff S. (2014) MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8694. Springer, Cham.
- [31] R. Tao, E. Gavves and A. W. M. Smeulders, "Siamese Instance Search for Tracking," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 1420-1429.
- [32] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Highspeed tracking with kernelized correlation filters. IEEE TPAMI, 37(3):583–596, 2015.
- [33] H. Fan and H. Ling, "Parallel Tracking and Verifying: A Framework for Real-Time and High Accuracy Visual Tracking," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 5487-5495.
- [34] Q. Yu, T.B. Dinh, and G. Medioni, "Online Tracking and Reacquisition Using Co-Trained Generative and Discriminative Trackers," Proc. 10th European Conf. Computer Vision, 2008.
- [35] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409-1422, July 2012.
- [36] Q. Hu, Y. Guo, Z. Lin, W. An and H. Cheng, "Object Tracking Using Multiple Features and Adaptive Model Updating," in IEEE Transactions on Instrumentation and Measurement, vol. 66, no. 11, pp. 2882-2897, Nov. 2017.
- [37] J. Li, X. Zhou, S. Chan and S. Chen, "Robust Object Tracking via Large Margin and Scale-Adaptive Correlation Filter," in IEEE Access, vol. 6, pp. 12642-12655, 2018.
- [38] J. S. Kim, D. H. Yeom and Y. H. Joo, "Fast and robust algorithm of tracking multiple moving objects for intelligent video surveillance systems," in IEEE Transactions on Consumer Electronics, vol. 57, no. 3, pp. 1165-1170, August 2011.
- [39] S. Zhao, S. Zhang and L. Zhang, "Towards Occlusion Handling: Object Tracking With Background Estimation," in IEEE Transactions on Cybernetics, vol. 48, no. 7, pp. 2086-2100, July 2018.
- [40] Prasanna Kumar Reddy Katuru, Vivek Josyula, Praveen Samudrala, M. Lokanath, P. Ashish, 2017, "Color Based Vehicle Detection and Tracking using Kalman Filter with a Fractional Feedback Loop", Advances in Computational Sciences and Technology (ACST), Volume 10, Year: 2017, Pages: 2565-2585, Research India Publications, ISSN: 0973-6107
- [41] Worrall A.D., Marslin R.F., Sullivan G.D., Baker K.D. (1991) Model-based Tracking. In: Mowforth P. (eds) BMVC91. Springer, London, https://doi.org/10.1007/978-1-4471-1921-0_39
- [42] Koller, D., Daniilidis, K. & Nagel, H.H. Int J Comput 11263 on (1993) 10: 257. <https://doi.org/10.1007/BF01539538>
- [43] KaewTraKulPong P., Bowden R. (2002) An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. In: Remagnino P., Jones G.A., Paragios N., Regazzoni C.S. (eds) Video-Based Surveillance Systems. Springer, Boston, MA, https://doi.org/10.1007/978-1-4615-0913-4_11
- [44] M. Yokoyama and T. Poggio, "A contour-based moving object detection and tracking," 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 271-276.
- [45] Kass, M., Witkin, A. & Terzopoulos, D. Int J Computer Vision (1988) 1: 321. <https://doi.org/10.1007/BF00133570>
- [46] Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-577, May 2003.
- [47] E Ali, EU Khan, EZ Mahmudi, R Ullah. "A Comparison of FAST, SURF, Eigen, Harris, and MSER Features." International Journal of Computer Engineering and Information Technology 8, no. 6 (2016): 100-105.
- [48] N N. Saunier and T. Sayed, "A feature-based tracking algorithm for vehicles in intersections," The 3rd Canadian Conference on Computer and Robot Vision (CRV'06), 2006, pp. 59-59.
- [49] V. Buddubarik, S. G. Tulluri and S. Mukherjee, "Multiple object tracking by improved KLT tracker over SURF features," 2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Patna, 2015, pp. 1-4.
- [50] E Ali, EU Khan, EZ Mahmudi, R Ullah. "A Comparison of FAST, SURF, Eigen, Harris, and MSER Features." International Journal of Computer Engineering and Information Technology 8, no. 6 (2016): 100-105.
- [51] H. Schweitzer, R. Deng and R. F. Anderson, "A Dual-Bound Algorithm for Very Fast and Exact Template Matching," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 3, pp. 459-470, March 2011.
- [52] S. Oron, A. Bar-Hillel, D. Levi and S. Avidan, "Locally Orderless Tracking," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 1940-1947.
- [53] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 1910-1917.
- [54] T. B. Dinh, N. Vo and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," CVPR 2011, Colorado Springs, CO, USA, 2011, pp. 1177-1184.
- [55] Y. Wu, J. Lim and M. Yang, "Object Tracking Benchmark," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1834-1848, 1 Sept. 2015.
- [56] S. D. Wei and S. H. Lei, "Fast template matching based on normalized cross correlation with adaptive multilevel winner update," IEEE Transactions on Image Processing, vol. 17 no. 11, pp. 2227-2235, 2008.
- [57] D. G. Lowe, "Distinctive Image Features from Scale Invariant Key-points," International Journal of Computer Vision, 2004, vol. 60, no. 2, pp. 91-110.
- [58] Calonder M., Lepetit V., Özuysal M., Trzcinski T., Strecha C. and Fua P., "BRIEF: Computing a Local Binary Descriptor Very Fast", IEEE Trans. Pattern Anal. Mach. Intell., 34(7):1281-1298, 2012.
- [59] Bay H., Tuytelaars T. and Gool L.V., "SURF: Speeded Up Robust Features", Comput. Vis. Image Understand., vol. 110, no. 3, pp. 346-359, 2008.
- [60] Jianbo Shi and C. Tomasi, "Good features to track," 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, 1994, pp. 593-600.
- [61] Tomasi, Carlo & Kanade, Takeo. (1991). Detection and Tracking of Point Features. International Journal of Computer Vision. 9. 137-154.
- [62] Hartley, R., & Zisserman, A. (2004). Multiple View Geometry in Computer Vision. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511811685
- [63] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-511-I-518 vol.1.
- [64] A. Opelt, A. Pinz, M. Fussenegger and P. Auer, "Generic object recognition with boosting," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 3, pp. 416-431, March 2006.
- [65] Greg Welch and Gary Bishop. 1995. An Introduction to the Kalman Filter. Technical Report. University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.
- [66] Y. Hua, K. Alahari and C. Schmid, "Online Object Tracking with Proposal Selection," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 3092-3100.



Rakesh Chandra Joshi

Rakesh Chandra Joshi was born in Champawat, India in 1994. He received the B.Tech. degree from Invertis University, Bareilly, India in 2015. He has completed the M.Tech. degree from Govind Ballabh Pant University of Agriculture and Technology, Pantnagar, India in 2018 and currently, he is pursuing independent research in area of image processing. His research interests include Digital

Image Processing, Machine Learning and Surveillance Systems.



Adithya Gaurav Singh

Adithya Gaurav Singh has completed the B.Tech. degree from Govind Ballabh Pant University of Agriculture and Technology, Pantnagar, India in 2018. Currently, he is associated with BaseApp Systems & Software Pvt. Ltd. His research interests include Computer Vision, Robotics and Image processing.



Mayank Joshi

Mayank Joshi has completed the B.Tech. degree from Govind Ballabh Pant University of Agriculture and Technology, Pantnagar, India in 2018 and currently, he is in the process of joining for higher studies. His research interests include Robotics, Computer Vision and Embedded Systems.



Sanjay Mathur

Sanjay Mathur received the B.E. degree from Madan Mohan Malaviya Engineering College, Gorakhpur, India in 1985, the M.E. degree from the University of Roorkee, presently IIT Roorkee, India in 1996, and the Ph.D. from Govind Ballabh Pant University of Agriculture and Technology, Pantnagar, India in 2009. Currently he is a Professor in the Department of Electronics and Communication

Engineering, Govind Ballabh Pant University of Agriculture and Technology, Pantnagar. His research interests include Communication Systems, Signal Processing, Image and Audio Processing, Neural Networks and Smart Antenna. He is a Fellow of Institution of Engineers (I) and Chairman of Pantnagar Local Centre. He is also a Life member of Indian Society of Technical Education (ISTE).