

RESEARCH ARTICLE

Multi-vehicle detection with identity awareness using cascade Adaboost and Adaptive Kalman filter for driver assistant system

Baofeng Wang, Zhiquan Qi*, Sizhong Chen, Zhaodu Liu, Guocheng Ma

Laboratory of Vehicle Engineering, School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

* qizhiquan@bit.edu.cn



OPEN ACCESS

Citation: Wang B, Qi Z, Chen S, Liu Z, Ma G (2017) Multi-vehicle detection with identity awareness using cascade Adaboost and Adaptive Kalman filter for driver assistant system. PLoS ONE 12(3): e0173424. <https://doi.org/10.1371/journal.pone.0173424>

Editor: Xiaosong Hu, Chongqing University, CHINA

Received: November 5, 2016

Accepted: February 19, 2017

Published: March 15, 2017

Copyright: © 2017 Wang et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: Testing files are available from <https://figshare.com/s/f51b64d177dfe9fac0f2/>.

Funding: This research is sponsored by the National Natural Science Foundation of China under Grant No. 51005019. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Vision-based vehicle detection is an important issue for advanced driver assistance systems. In this paper, we presented an improved multi-vehicle detection and tracking method using cascade Adaboost and Adaptive Kalman filter(AKF) with target identity awareness. A cascade Adaboost classifier using Haar-like features was built for vehicle detection, followed by a more comprehensive verification process which could refine the vehicle hypothesis in terms of both location and dimension. In vehicle tracking, each vehicle was tracked with independent identity by an Adaptive Kalman filter in collaboration with a data association approach. The AKF adaptively adjusted the measurement and process noise covariance through on-line stochastic modelling to compensate the dynamics changes. The data association correctly assigned different detections with tracks using global nearest neighbour(GNN) algorithm while considering the local validation. During tracking, a temporal context based track management was proposed to decide whether to initiate, maintain or terminate the tracks of different objects, thus suppressing the sparse false alarms and compensating the temporary detection failures. Finally, the proposed method was tested on various challenging real roads, and the experimental results showed that the vehicle detection performance was greatly improved with higher accuracy and robustness.

Introduction

Advanced driver assistance systems for cars like forward collision warning(FCW), autonomous emergency braking(AEB) and adaptive cruise control(ACC), require a reliable perception of the preceding vehicles. Different sensors including radar, Lidar, stereo vision and monocular vision have been researched for vehicle detection. With the availability of low-cost-high-performance cameras and feasible computer vision technologies, vision based vehicle detection methods are becoming an active area of research in recent years [1]. In this paper, we focus on the research of monocular-based multi-vehicle detection.

Robust vehicle detection using monocular vision is a challenging problem. Traditional methods tended to use priori knowledge of vehicle rears such as edge [2], shadow [3], and symmetry [4] to hypothesized the location of the vehicles in the image plane. These methods has been verified to be effective and fast in simple situations, but cannot achieve a satisfying detection under complex scenarios such as pose change, illumination variation, noisy background, etc. Motion cues such as optical flow have also been used for vehicle detection [5]. But these methods require a relatively stable background, and are more suitable for traffic surveillance with static cameras or stereo-vision based method. With the development of machine learning and data mining technology, the machine learning based methods are becoming more and more popular in different research field [6–8]. For vision based vehicle detection, modern methods often employ different classification models including Support Vector Machine(SVM), Adaboost, deep neural networks etc., with higher level local features, such as Histogram of Oriented Gradient(HOG), Haar-like features and Convolutional Neural Networks features, thus to locate the vehicles more robustly under various challenging conditions.

However, like all the other detectors, the measurements given by vision-based algorithm also come with occasional detection failures and false alarms due to various environments. To improve the detection performance, different approaches have been proposed. Trivedi et al. [10] developed a vehicle detector based on Adaboost classification and Haar-like features, and proposed an active-learning framework to improve the discriminating capacity of the classifier through iterative training with the ambiguous misclassified samples. Stanculescu et al. [11] improved the classifier through introducing new Adaboost features that extended by the topological properties of vehicle rear appearance. Kim [12] proposed Haar-like edge features to work in cooperation with the traditional Haar-like intensity features, and efficiently reduced the false detections. Context of the driving scene is another efficient way to improve the detection accuracy, and among which the vanishing point(VP) is the most popular cue to filter out the vehicle hypotheses with locations above it [13]. But the limitation of this approach is that the false alarms below VP cannot be filtered out.

The aforementioned methods in general focus mainly on improving detection performance on a single frame, while further improvement could be achieved by a tracking mechanism through filtering from multiple frames. Vehicle tracking has been addressed by many researchers, different methods such as Kalman filter(KF), extended Kalman filter(EKF) and particle filter(PF) have been applied, and KF is especially popular due to its simplicity and effectiveness [4, 12]. Generally, KF were used to estimate the possible region of interest(ROI) for vehicle redetection [4], compensate the temporary detection losses, and smooth the jittery trajectory [12]. However in real world vehicle tracking scenario, there are two limitations of conventional KF. First, conventional KF requires a precise prior knowledge of the the process noise and measurement noise properties(i.e. the matrices Q and R , respectively) which are usually pre-set and remain constant throughout the whole tracking process [14, 15]. Because of changing distance and scale when approaching or departing the front vehicles, the noise properties of vehicle objects vary greatly, which will seriously degrade the filtering performance and may even result in filter divergence during long-term tracking [16]. Secondly, aforementioned approaches mainly focused on how to predict and update the state of the target vehicles with measurements, but didn't discuss how to resolve the identity assigning problem of different objects across frames. In fact, scenarios with multi-vehicle are quite common in real road driving, and the identity awareness between observations and measurements is the foremost issue for a multi-object filtering mechanism. Additionally, the changing number of vehicle targets in the field of view(FOV) as well as the occasional detection misses and sparse false alarms make the association procedure more challenging.

Based on the points discussed above, we presented an improved multi-vehicle detection and tracking method using cascade Adaboost and adaptive Kalman filter(AKF) with target identity awareness in this paper. The outline of the proposed method is shown in Fig 1. Vehicles are detected by a cascade Adaboost classifier using Haar-like features, followed by a more comprehensive verification process which could refine the vehicle hypothesis in terms of both location and dimension. In vehicle tracking, each vehicle is tracked with identity awareness by AKF in collaboration with a data association approach. The AKF adapts the system noise property through on-line stochastic modelling, the data association correctly assigns different measurements with observations using global nearest neighbour(GNN) algorithm while considering the local validation. During tracking, a temporal context based track management is proposed to decide whether to initiate, maintain or terminate the tracks of different objects. Finally, the proposed method was tested on various challenging real roads, and the experimental results showed that the vehicle detection performance was greatly improved.

The remainder of this paper is organized as follows: the section “Cascade Adaboost for vehicle detection” details the vehicle hypothesis generation and verification method. In the section

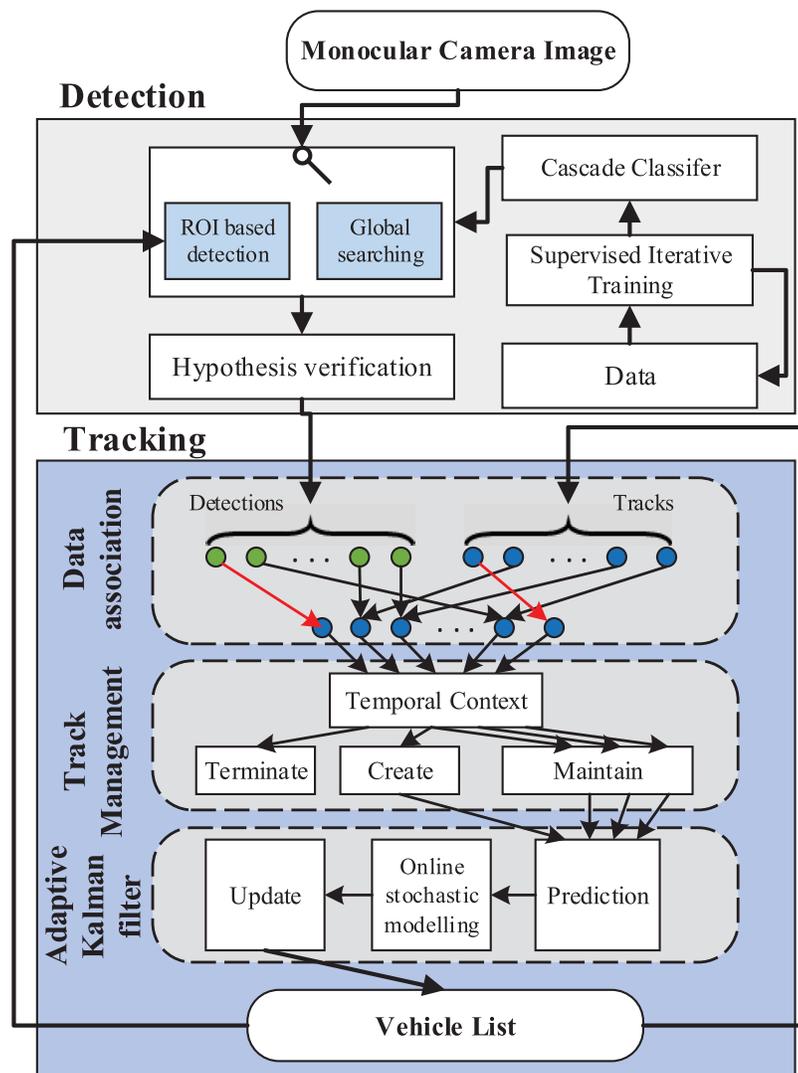


Fig 1. Flow chart of multi-vehicle detection and tracking.

<https://doi.org/10.1371/journal.pone.0173424.g001>

“Multi-vehicle tracking with identity awareness”, the adaptive Kalman filter with on-line stochastic modelling, the data association method for identity assignment, and the track management method are presented. The experiment results and system performance evaluation of different scenarios are presented in “Experiments”. Conclusions and future works are discussed in “Conclusion”.

Cascade Adaboost for vehicle detection

Cascade classifier based on Haar-like features

Haar-like features based boosted cascade was first introduced by Viola and Jones for face detection [9]. As shown in Fig 2, Haar-like features are a set of different combinations of adjacent rectangles with various scales, positions and aspect ratios, whose value is the difference of sum of intensities between the black area and white area, given by

$$f = \sum_{(m,n) \in R_w} I(m, n) - \sum_{(m,n) \in R_b} I(m, n) \tag{1}$$

where R_w and R_b represent white area and black area respectively. This value indicates certain characteristics of the image area such as edge, textures and symmetry, thus could be used to encode the local appearance of objects. Given the fact that the appearance of the vehicle rear typically maintains some consistent edges, rectangles, shadows and symmetries as shown in Fig 2, Haar-like features is well-suited for vehicle appearance description.

Let $F \propto \{f_1, \dots, f_N\}$ represent the Haar-like feature family, and \mathbb{S} the training set with N_s training samples. In the training process, each image sample of the training set is represented by Haar-like feature family $\mathbb{S} \propto \{(F_i, L_i) | i \in \{1, 2, \dots, N_s\}, L_i \in \pm 1\}$, where L_i is the sample label with 1-vehicle and -1-non-vehicle. The extracted features are selected as weak filters by Adaboost through linear combination and weighting to generate a stronger classifier which minimizing the classification error,

$$\hat{c}_w = \arg \min \sum_{i=1}^{N_s} c_w(F_i) \tag{2}$$

Given an image patch with Haar features of F_i , if $\hat{c}_w(F_i) > 0$ it will be classified as vehicle by this classifier. Through iterative training, a cascade of Adaboost classifiers are generated and structured stage by stage into a binary decision tree as shown in Fig 3. The image patches surviving the last classifier stage will be taken as vehicle hypotheses. For efficiency, the less accurate but faster classifiers are arranged in earlier stages, so as to reject the obvious non-vehicles

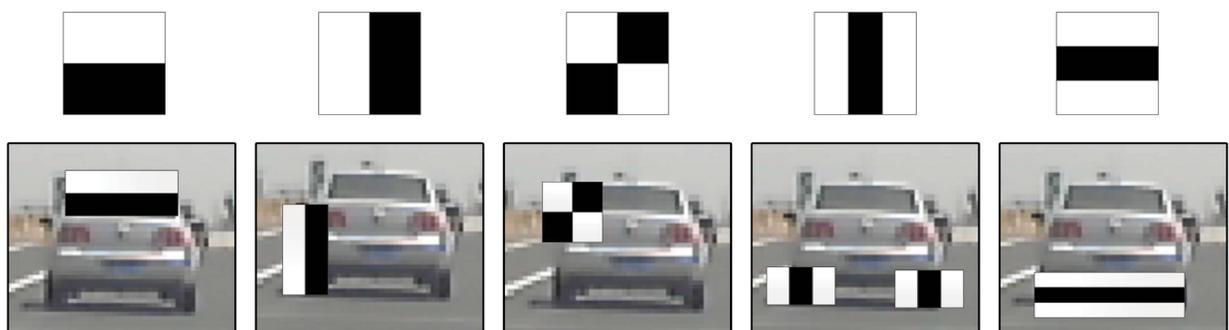


Fig 2. Haar-like features. Top row: basic forms of Haar-like features. Bottom row: vehicle rear appearances suitable for Haar-like features.

<https://doi.org/10.1371/journal.pone.0173424.g002>

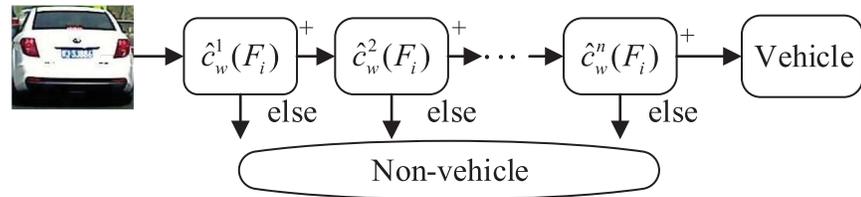


Fig 3. Cascade Adaboost classifier for vehicle detection.

<https://doi.org/10.1371/journal.pone.0173424.g003>

with little processing. The confidence of a vehicle hypothesis could be defined as

$$C_w(F_i) = \sum_{j=1}^n \hat{c}_w^j(F_i) \tag{3}$$

where n is the number of the stages of the cascade classifier.

Dataset collection and classifier training

Training samples are generally collected by manual cropping, which is a very labour-cost procedure. In data collection for classifier training, we adopted a coarse to fine rule. First, a certain amount of samples were manually cropped to hatch a initial classifier. Afterwards, the classifier was run on new data to generate more samples for retraining. Compared with manually cropping the raw image, the detection results are more easily to select and process, thus a massive samples could be collected with less time and labour cost. The collected sample were used for retraining to generate a more advanced classifier. Through fast iterative training on booming data under the supervision of people, the performance of the classifier could be improved rapidly. In this case, we carefully cropped 2881 positive samples and 6650 negative samples for initial training and normalized them into 20×20 form. After 4 rounds of iterative training, a cascade vehicle detector with 20 stages was generated after training on 23641 positive samples and 41562 negative samples in total. Some of the training samples are as shown in Fig 4.



Fig 4. Training samples for classifier.

<https://doi.org/10.1371/journal.pone.0173424.g004>

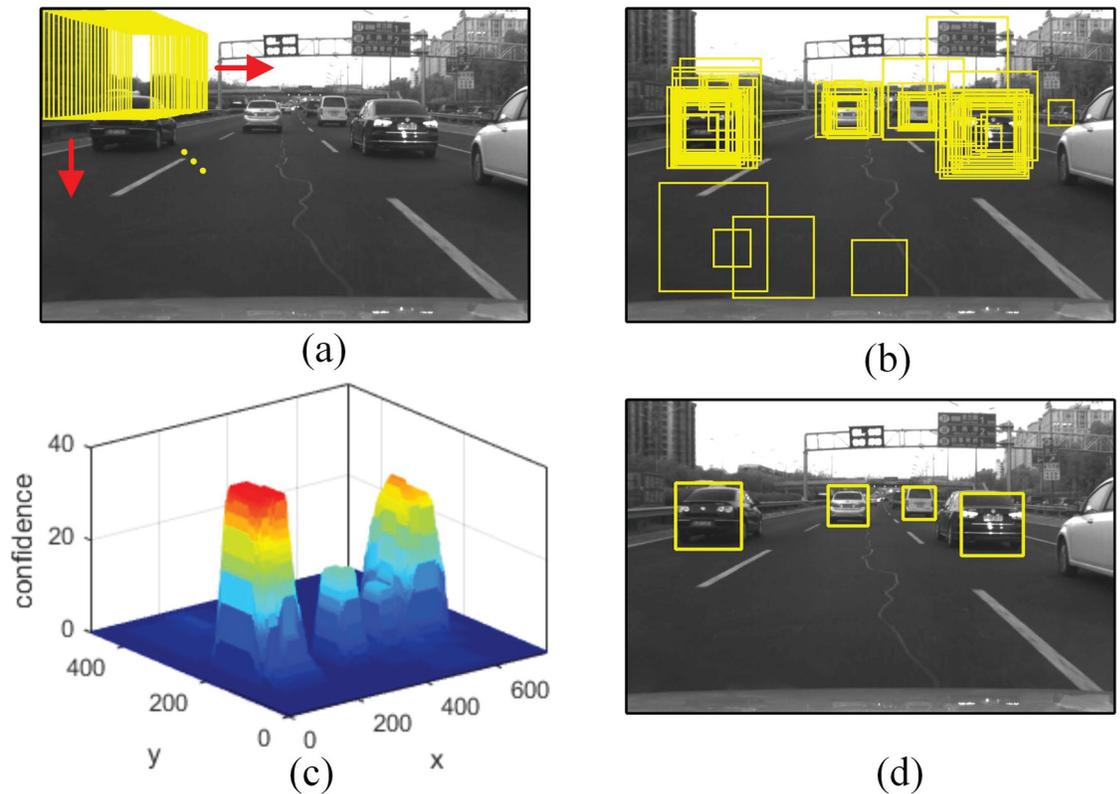


Fig 5. Vehicle detection scheme. (a) Sliding window sampling. (b) Raw detection results. (c) Confidence map of raw detections. (d) Final results after redundant boxes fusion.

<https://doi.org/10.1371/journal.pone.0173424.g005>

Hypothesis generation

In vehicle hypothesis generation, cascade classifier adopts a sliding window approach. As shown in Fig 5(a), the window scans the image globally at all scales and locations to sample patches which are then checked by the classifier cascade. Since the classifier tolerates a wider region misalignment, several redundant windows near the ideal location of a vehicle object will be returned with different confidences, as shown in Fig 5(b) and 5(c). The raw detection results are then grouped by clustering and non-maxima suppression method [17], and finally vehicle hypotheses are generated in the form of squared bounding boxes as shown in Fig 5(d), which could be given by,

$$v_i = [x_i \ y_i \ w_i], \ i = 1, 2, 3, \dots \quad (4)$$

where i is the index of the object in the list, (x_i, y_i) is defined as the center of bottom line of the bounding box, and w_i is the side length of the bounding box.

The global sliding window approach is too time-consuming for real-time application. To improve the processing speed, we adopted the ROI based method to narrow down the search region according to the detection results of last frame. To detect the new appearing targets in time, the global searching is applied periodically every several frames. The period of global searching is experimentally set as 5 frames in this paper. Considering the average processing speed of our method (discussed in the Experiments section), the driving scene will be globally

scanned about every 230ms, which is very reliable for the ADAS system to find new appearing vehicles in time.

Hypothesis refinement

Raw vehicle hypotheses are usually mixed with some false detections, therefore a post-processing is very essential to verify the hypotheses. It is a priori knowledge that the widths of different vehicles (from cars to truck) are within a certain range (about 1.5m–2.5m), and all the vehicles are located on the road. Therefore the raw hypotheses that violate this constraint could be safely filtered out as false detections. To get the real width of a vehicle hypothesis, the projection model of camera is used,

$$\begin{bmatrix} x_I \\ y_I \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_{cu} & \alpha f_{cu} & c_u \\ 0 & f_{cv} & c_v \\ 0 & 0 & 1 \end{bmatrix}}_K \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{R \quad T} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{5}$$

where $[x_I, y_I]^T$ is the image coordinates, $[x_w, y_w, z_w]^T$ is the corresponding real world location K is the intrinsic matrix with (f_{cu}, f_{cv}) the focal lengths expressed in pixels units vertically and horizontally, (c_u, c_v) the principal point, and α the skew coefficient. R and T are the extrinsic parameters defining the rotation and translation from world coordinate to camera coordinate. These parameters could be calibrated once the camera is installed in the vehicle. Theoretically, the projection transformation is irreversible due to the loss of depth information, however based on the flat road assumption [18], Eq 5 could be simplified as

$$\begin{bmatrix} x_I \\ y_I \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_{cu} & \alpha f_{cu} & c_u \\ 0 & f_{cv} & c_v \\ 0 & 0 & 1 \end{bmatrix}}_M \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \tag{6}$$

This simplified projection model is reversible, and given a vehicle hypothesis $v_i = [x_i, y_i, w_i]$, its width in real world could be estimated,

$$\bar{w}_w = \left| M^{-1} \left[x_i - \frac{w_i}{2}, y_i, 1 \right] - M^{-1} \left[x_i + \frac{w_i}{2}, y_i, 1 \right] \right| \tag{7}$$

Eq 7 shows that the estimated width \bar{w}_w takes into consideration of both the location and dimension of the vehicle hypothesis. According to the prior knowledge, only the vehicle hypotheses with proper location and dimension will have a estimated width within the reasonable range, while the others violating this constraint could be safely eliminated as false alarms. In this paper, we define the width range as $[w^-, w^+]$, where w^- and w^+ are the predefined upper and the lower width threshold of the vehicle. To account for the measurement uncertainty, we relax the width range of vehicle as $[1.2, 3]m$ through experiments. Compared with traditionally used vanishing point for vehicle hypothesis refinement, this constraint could provide a more comprehensive verification and filter out more false detections, see Fig 6 for example.

Multi-vehicle tracking with identity awareness

In this section, the raw measurements detected by the vehicle detector are filtered by a multi-object tracking mechanism. In this tracking mechanism, the AKF with on-line stochastic



Fig 6. Hypothesis refinement result. (Left) Vanishing point based method fails to filter the false hypotheses with bottom edge under vanish point; (Right) Proposed method successfully filters out the false hypothesis.

<https://doi.org/10.1371/journal.pone.0173424.g006>

modelling is developed for the state filtering of each object, a GNN based approach with local validation is used for data association, and a temporal context based track management is built to create, maintain, or terminate the tracking each object based on the temporal context of their detection histories.

Adaptive Kalman filter

With the detection result given by the detector, we define the state vector of a vehicle object $v_i(k)$ at the discrete time instant k as

$$X_i(k) = [x_i(k), y_i(k), w_i(k), \dot{x}_i(k), \dot{y}_i(k), \dot{w}_i(k)] \tag{8}$$

where the definitions of the parameters are the same as in Eq 4. Since the time intervals between two successive frames are very small due to the high acquisition frequency of camera, the inter-frame state change of moving targets could be considered as uniform motion. Thus the system and measurement models of the vehicle are defined as follows:

$$X_i(k) = AX_i(k-1) + \omega_i(k-1), \omega_i(k-1) \sim N(0, Q_{k-1}) \tag{9}$$

$$Z_i(k) = CX_i(k) + v_i(k), v_i(k) \sim N(0, R_k) \tag{10}$$

where $\omega_i(k)$ is the white Gaussian process noise with covariance of $Q(k-1)$, $v_i(k)$ is the white Gaussian measurement noise with the covariance of $R(k)$, A and C are the state transition matrix and measurement matrix, which could be defined respectively as,

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

where Δt is the time of the frame interval.

As mentioned above, the KF pre-sets the process noise and measurement noise, neglecting the noise level change which is very common in vision based vehicle tracking due to changing distance and scale. When the stochastic properties of the process and measurement noises becomes improper, large bias or oscillation will occur. To remedy the effect caused by changing noise properties, we built an AKF filter which could adapt the properties of measure and system noise covariance in on-line manners. The proposed filter is composed of three steps: prediction, on-line stochastic modelling and update.

Prediction step. The priori state and the state covariance of the vehicle in the following frame is predicted,

$$X_i(k|k-1) = AX_i(k-1|k-1) \tag{13}$$

$$P_i(k|k-1) = AP_i(k-1|k-1)A^T + Q_i(k-1) \tag{14}$$

where $X_i(k|k-1)$ and $P_i(k|k-1)$ denote the predicted state vector and innovation covariance of next epoch respectively, $X_i(k-1|k-1)$ and $P_i(k-1|k-1)$ are the filtered state vector of last epoch.

On-line stochastic modelling. Before updating the system directly, we modelled the properties of the measurement noise based on the innovation sequence which could be defined as

$$\begin{aligned} \varepsilon_i(k) &= Z_i(k) - CX_i(k|k-1) = CX_i(k) + v_i(k) - CX_i(k|k-1) \\ &= C[X_i(k) - X_i(k|k-1)] + v_i(k) \end{aligned} \tag{15}$$

where $Z_i(k)$ is the real measurement. It could be seen that the innovation reflects the difference between the prediction and the real measurements, and is a zero-mean Gaussian with noise sequence. Taking variances on both sides, we could obtain the innovation covariance as,

$$\begin{aligned} C_{\varepsilon_i}(k) &= E[\varepsilon_i(k)\varepsilon_i(k)^T] = CP_i(k|k-1)C^T + R_k \\ &= C[AP_i(k-1|k-1)A^T + Q_i(k-1)]C^T + R_k \end{aligned} \tag{16}$$

To get some statistical smoothing, the calculation of covariance is carried out through averaging the innovation samples inside a moving window at epoch k ,

$$C_{\varepsilon_i}(k) = \frac{1}{W_m} \sum_{i=0}^{W_m-1} \varepsilon_i(k-i)\varepsilon_i(k-i)^T \tag{17}$$

where W_m is the size of the window. From Eq 12 to Eq 14, the measurement covariance could be adaptively estimated as

$$R_i(k) = C_{\varepsilon_i}(k) - CP_i(k|k-1)C^T \tag{18}$$

Update step. With the estimated measurement covariance, a posteriori state is estimated based on Bayes' law in the update step as follows,

$$K_i(k) = P_i(k|k-1)C^T[CP_i(k|k-1)C^T + R_i(k)]^{-1} \tag{19}$$

$$X_i(k|k) = X_i(k|k-1) + K_i(k) [Z_i(k) - CX_i(k|k-1)] \tag{20}$$

$$P_i(k|k) = [I - K_i(k) C] P_i(k|k-1) \tag{21}$$

where $K_i(k)$ is the Kalman gain, which defines the updating weight between the new measurement and the prediction from the system dynamic model. Based on the residual between the measurement and filtered states, the process noise $Q_i(k)$ is also estimated for updating,

$$Q_i(k) = \frac{1}{m} \sum_{i=0}^{m-1} \Delta X(k-i) \Delta X(k-i)^T + P_i(k|k) - AP_i(k-1|k-1)A^T \tag{22}$$

where $\Delta X(k-i) = X_i(k) - X_i(k|k-1)$. With the on-line stochastic modelling, the properties of the process noise and the measurement noise used in the filter are adaptively updated so as to avoid the filtering divergence caused by changing dynamics during vehicle tracking.

Data association

In multiple objects tracking, the problem before filtering is to decide the correct identity association between the observations and measurements. Different methods such as nearest-neighbouring(NN) [19], jointed probability data association(JPDFAF) [20], and multiple hypotheses tracker(MHT) [21] have been proposed for multi-target association. Considering the efficiency and reliability, we adopted the global nearest neighbour(GNN) [22] approach for data association while considering the local validity.

The data association problem could be formulated as a linear assignment problem which searches one-to-one mapping solution between the detection list of m measurements and the track list of n tracked objects,

$$\hat{H}(k) = \arg \max \sum_{j=1}^m \sum_{i=1}^n l_k(i,j) \bullet h_k(i,j)$$

$$H(k) = \begin{bmatrix} h_k(1,1) & h_k(1,2) & \dots & h_k(1,m) \\ h_k(2,1) & h_k(2,2) & \dots & h_k(2,m) \\ \vdots & \vdots & \dots & \vdots \\ h_k(n,1) & h_k(n,2) & \dots & h_k(n,m) \end{bmatrix}$$

$$s.t. \begin{cases} \sum_{i=1}^n h_k(i,j) \in \{0,1\}, \\ \sum_{j=1}^m h_k(i,j) \in \{0,1\}, \\ h_k(i,j) \in \{0,1\}, \\ h_k(i,j) = 0, j \in \{0,1,\dots,m\}, \text{ if } \sum_{i=1}^n l_k(i,j) = 0, \\ h_k(i,j) = 0, i \in \{0,1,\dots,n\}, \text{ if } \sum_{j=1}^m l_k(i,j) = 0, \end{cases} \tag{23}$$

where H is the mapping matrix.

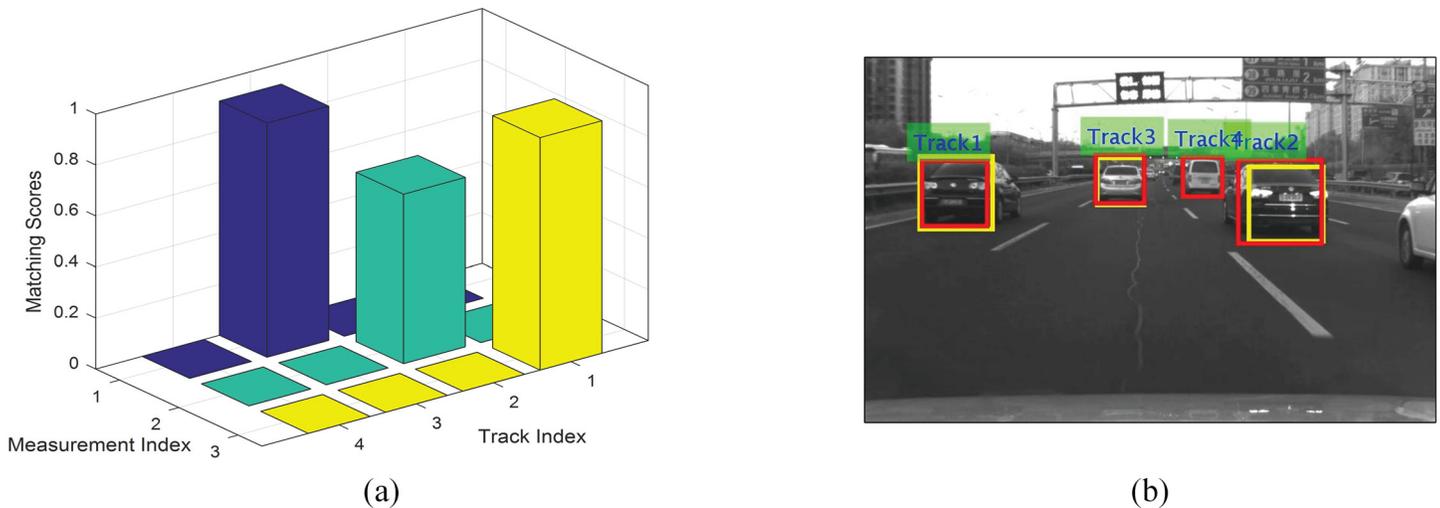


Fig 7. Data association result. (Left) The visualization of the matching confidence matrix. (Right) Association results between the measurements (yellow) and existing tracks (red).

<https://doi.org/10.1371/journal.pone.0173424.g007>

The selection of likelihood should try to maximize the matching degree of correct association hypotheses, while suppressing those of the false ones. Traditional methods usually use the distances (Mahalanobis or Euclidean) of the observation states for evaluation [23]. However this method needs a set of scalars to balance the weights between different state variables (location and dimension), which is not easy to tune. Some more advanced method also used high-dimensional appearance features through on-line learning to handle more challenging scenarios such as hundreds of crowded objects tracking with high occlusions [24]. But these methods seem too luxurious in our case, since preceding vehicles in ADAS scenarios are generally box-shaped, and the vehicles involved in occlusion scenario usually have significant scale differences in 2D image plane (due to their different longitudinal distances) which is very reliable for identity discrimination in cooperation with location. Overlap ratio can measure the similarity of two objects in terms of both the location and dimension without weighting, thus is selected for the matching confidence evaluation in this paper. It is worth to mention that the GNN method does not account for the local validity of one particular association hypothesis, and wrong association result may be generated when partial occlusion between different vehicles exists. To avoid this problem, an independent threshold is imposed to check the local validity of each association hypothesis, therefore the likelihood is defined as follows:

$$l_k(i, j) = \begin{cases} \frac{\tilde{v}_j(k) \cap \hat{v}_i(k)}{\tilde{v}_j(k) \cup \hat{v}_i(k)}, & \text{if } \frac{\tilde{v}_j(k) \cap \hat{v}_i(k)}{\tilde{v}_j(k) \cup \hat{v}_i(k)} > l_t \\ 0, & \text{else.} \end{cases} \quad (24)$$

where l_t is the threshold for local validation. A typical association example is as shown in Fig 7, there are four existing tracks while only three measurements are returned, with the proposed method, the measurements are corrected assigned with corresponding tracks for update.

Temporal context based track management

Multi-vehicle tracking in real road is a dynamical scene where vehicles are disappearing, appearing in the field of view, and the sparse false detections and detection failures bring it

more uncertainty. To track the vehicles more reliably, we have developed a track management approach to decide whether to initiate, maintain or terminate tracks according to the temporal context of their tracking histories.

In our management system, the detection history of each vehicle is observed simultaneously in the track level,

$$T_i(k) = [T_i^{C_+}(k), T_i^{C_-}(k), T_i^S(k)] \tag{25}$$

where $T_i^{C_+}(k)$ is the count of consecutive frames with successful detections, $T_i^{C_-}(k)$ is the count of consecutive frames with detection failures, and $T_i^S(k)$ is the track state of the target. In this paper, we defined four track states: $T_i^S(k) \in \{0, 1, 2, 3\}$, where $T_i^S(k) = 0$ means terminated track, $T_i^S(k) = 1$ hypothesized track, $T_i^S(k) = 2$ registered track, and $T_i^S(k) = 3$ decaying track. The state transition of each object is driven by the temporal context of their tracking history, which is given by,

$$T_i^S(k) = \begin{cases} 0, & C_-^T \leq T_i^{C_-}(k) \\ 1, & 0 < T_i^{C_+}(k) < C_+^T \\ 2, & C_+^T \leq T_i^{C_+}(k) \\ 3, & 0 < T_i^{C_-}(k) < C_-^T \end{cases} \tag{26}$$

where C_+^T is the threshold of consecutive successful count which is generally responsible for the track creation and maintenance, and C_-^T is the the threshold of consecutive failure count which is responsible for track termination. The selection of these thresholds depend on the detection performance(detection rate and false positive rate) of the vehicle detector. An advanced detector could afford more aggressive performance with lower thresholds. While the detector with lower performance may need higher thresholds to suppress the false alarms and compensate the frequent detection failures. In return, the selection of the thresholds also affects the the tracking performance. A higher C_+^T leads to a lower false alarm rate but a longer delay in the creation of a new track, and a higher C_-^T will produce a more continuous trajectory but a delayed track termination. The generic state transitions and following filtering operation are as shown in Fig 8. With the proposed track management, the false detections which usually occurs temporarily could be filtered out, and the wrong termination of a tracked vehicle due to momentary unsuccessful detection in one or several frames can also be avoided.

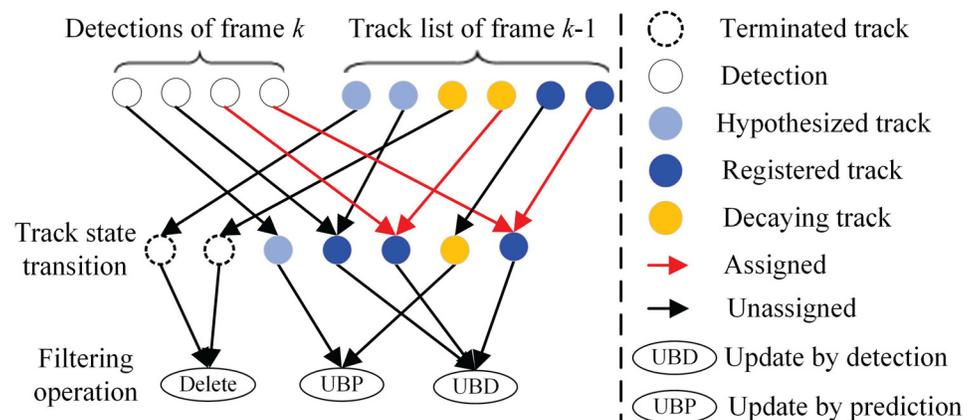


Fig 8. The generic track state transitions in the track management.

<https://doi.org/10.1371/journal.pone.0173424.g008>

Experiments

The proposed method was realized by C++/OpenCV and run on a Laptop with Pentium i5–2.5GHz, 2G RAM configuration. A large set of videos with form of $752 \times 480 \times 8$ have been captured under various real road scenarios in Beijing for the test. The performance of vehicle detection is first analysed. Then the tracking performance improvement introduced by AKF is presented through comparison with KF. Finally the multi-vehicle tracking performance in dynamical environment is evaluated.

Vehicle detection results

To verify the detection performance of the proposed method, 4 sets of images were chosen as test material. These dataset covers various types of vehicles including sedan, truck, coach, van, and SUV under different scenarios such illumination variation (dark and bright), scale change (large and small), and heavy traffic. Some vehicles with strong pose change and partial occlusion are also covered. Typical detection results of different scenarios are as shown in Fig 9. Table 1 shows the quantitative results of different datasets. The detection results were analysed in terms of true positive rate (TPR), false positive rate (FPR), and false negative rate (FNR). To show the improvement of the proposed hypotheses verification method, the vehicle detector using vanishing point (VDVP) [13] for hypothesis verification is applied for comparison. It is worth to note that vehicles that were too small (smaller than 30×30 pixel) and highly occluded (more than half of the rear appearance) were ignored. It could be seen that compared with



Fig 9. Detection results in different conditions. (a) Dark illumination. (b) Bright illumination. (c) Large scale due to ultra-near distance. (d) Small scale due to long distance. (e) Mottled shadow. (f-g) Slight partial occlusion. (h) Dense traffic. (i) Truck. (j) Van. (k) Strong pose change. (l) Clutter background.

<https://doi.org/10.1371/journal.pone.0173424.g009>

Table 1. Vehicle detection performance.

Dataset	Scenario	NOV	Ours			VDVP [13]		
			TPR	FPR	FNR	TPR	FPR	FNR
Dataset 1	Dark illumination	1846	94.8%	5.2%	2.4%	86.8%	13.2%	2.4%
Dataset 2	Bright illumination	2537	95.2%	4.8%	3.1%	88.1%	11.9%	3.1%
Dataset 3	Scale change	1748	95.3%	4.7%	6.2%	87.9%	12.1%	6.1%
Dataset 4	Heavy traffic	2215	94.7%	5.3%	2.9%	84.4%	15.6%	2.9%
Average		8346	95.0%	5.0%	3.5%	86.8%	13.2%	3.5%

<https://doi.org/10.1371/journal.pone.0173424.t001>

VDVP the proposed method could achieve a higher TPR(95.0%) with about the same FNR (3.5%), which means the proposed hypothesis verification method is more efficient to eliminate false alarms. Another thing needs to point is that the FNR of scale change scenario(6.2%) is relatively higher than that of others. It is because the rear appearance features get ambiguous with increasing distance, thus becomes more uncertain for the detector to discriminate.

Performance of AKF based vehicle tracker

To verify the filtering performance improvement introduced by the on-line stochastic modeling of AKF, a typical video sequence of a single vehicle object was filmed. In this sequence, the scale and the location of the target varies due to changing distance and curve lane, see Fig 10. To show the performance improvement, the conventional KF based vehicle tracker used in [12] was used as subject for comparison. The location error and width error are used as comparison metrics, which are respectively defined as

$$location_error = \sqrt{[\hat{x}(k) - x_{GT}(k)]^2 + [\hat{y}(k) - y_{GT}(k)]^2} \tag{27}$$

$$width_error = \hat{w}(k) - w_{GT}(k) \tag{28}$$

where $[\hat{x}(k), \hat{y}(k), \hat{w}(k)]$ represents the tracked vehicle in the k th frame, and $[x_{GT}(k), y_{GT}(k), w_{GT}(k)]$ the ground truth(GT) which has been manually labelled.

The tracking errors are as plotted in Fig 11. The maxima, mean and root-mean squares of these errors are shown in Table 2. It could be seen that as the tracking procedure proceeds, the tracking error of KF propagated greatly, while that of the proposed method were efficiently suppressed. It is because the noise property of the measurements changed due to scale and distance variations, and the pre-set process covariance and noise covariance of KF became appropriate. However, with the on-line stochastic modelling, the proposed method could adaptively estimate the covariance of process noise and measurement noise, thus compensate the noise properties changes, and avoid the tracking error propagation.

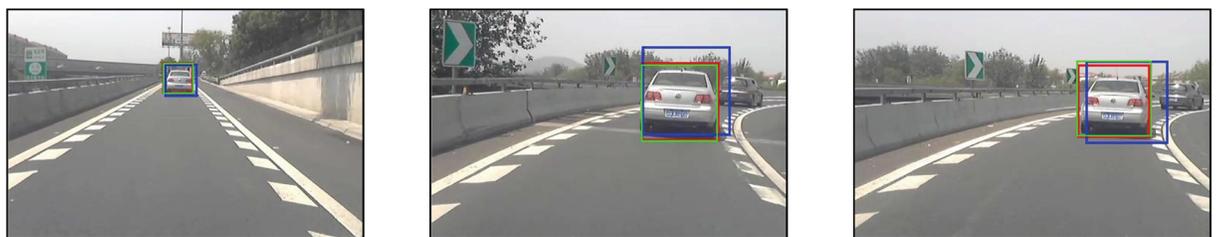


Fig 10. Video sequence of single vehicle with large scale change. Blue-KF, green-GT, red-ours.

<https://doi.org/10.1371/journal.pone.0173424.g010>

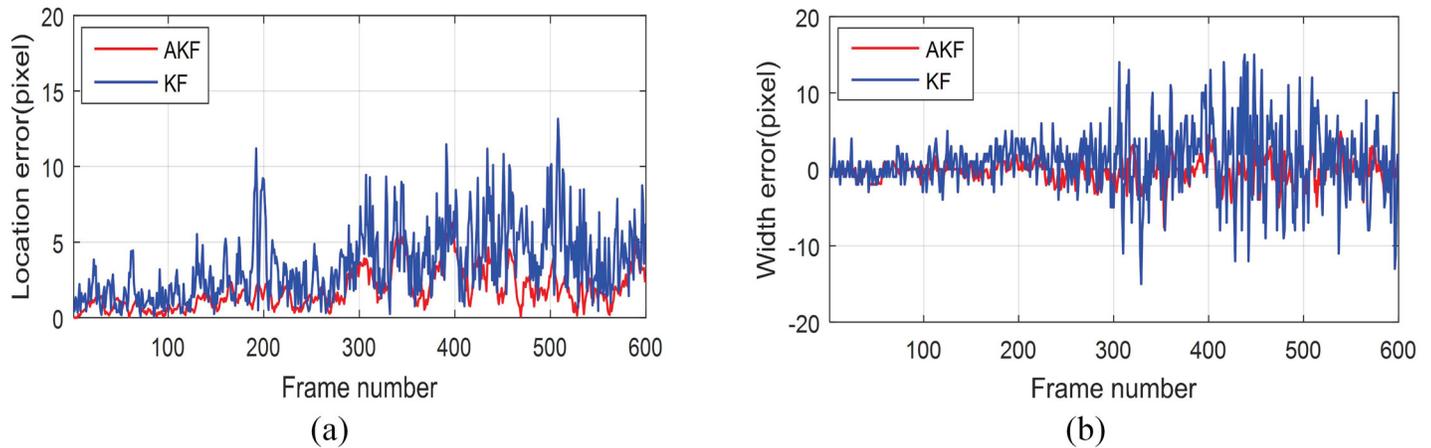


Fig 11. Tracking performance comparison between KF and AKF. (a)Tracking error of location. (b)Tracking error of width.

<https://doi.org/10.1371/journal.pone.0173424.g011>

Multi-vehicle tracking performance

To evaluate the multi-vehicle detection and tracking performance systematically, two video clips of typical driving scenarios with dense traffic were tested in this section. These video were captured on the 3-Ring and the 5-Ring road Beijing, which are busy urban express-ways with varying number of vehicles which were appearing, disappearing and reappearing in the field of view(FOV) due to frequent manoeuvres such as cut-in, lane change and overtaking. To better show the improvement of the proposed tracking method, the original vehicle detector with global searching schema was applied for comparison. Considering the detection performance of our detector, the thresholds in the track management were experimentally set as $C_+^T = 5$ and $C_-^T = 4$.

The video of 3-Ring road is composed of 1271 frames, and there were 6 different preceding vehicles across three lanes appeared during the tracking procedure. Some of the tracking examples are shown in Fig 12, the tracked trajectories of different vehicle targets are shown in Fig 13, and the full result could be found in the S1 File. It shows that our method could generally track all the relevant vehicle targets continuously and smoothly with correct identities throughout the whole video. Although Track# 1 and Track# 3 were cut by the aggressive lane change manoeuvre of Track# 6, their trajectories were successfully rebuilt by Track#7 and Track#8 after they reappeared. It is worth to note that there was a wrong identity switch from Track# 7 to Track# 9(see Frame 964–1068 in Figs 12 and 13(a)), it is because this target was too small due to the long distance from the ego vehicle, and a short-term detection failures occurs.

Figs 14 and 15 show the test results of the data captured on 5-Ring road. This video contains 1689 frames, and is featured by frequent overtaking vehicles appearing and disappearing in the

Table 2. Tracking error comparison between AKF and KF.

	AKF based tracker		KF based tracker [12]	
	Location	Width	Location	Width
MaxErr (pixel)	6.3	7.9	13.6	15.1
MeanErr (pixel)	1.8	1.2	3.5	3.1
RMS (pixel)	2.2	1.6	4.3	4.3

<https://doi.org/10.1371/journal.pone.0173424.t002>



Fig 12. Typical tracking samples of the video captured on 3-Ring road. (Yellow) Original cascade Adaboost detector. (Red) Ours. Note the detection failure and false alarm of original detector in Frame 61 and 738 respectively.

<https://doi.org/10.1371/journal.pone.0173424.g012>

FOV(Full result refers to [S1 File](#)). There are in total 13 vehicles shown up during the procedure. Sun glares and mottled shadows make the detection more challenging. Comprehensive analysis shows that occasional detection failures and false alarms also occurred in the detection results of the original vehicle detector. And the detection failures are especially severe since vehicles departed(see Frame 428, 616 1227 and 1689 in [Fig 14](#)) frequently. However these detection loss and false alarms were also efficiently compensated and eliminated by our method, and more continuous and smooth trajectories with correct identities were generated for each target throughout the sequences. It is worth to note that a few frames of false alarms also occurred in our method(0.2%), it was because of the termination delay caused by tracking mechanism that the object already disappeared due to occlusion, but the track mechanism still held the track for a few frames.

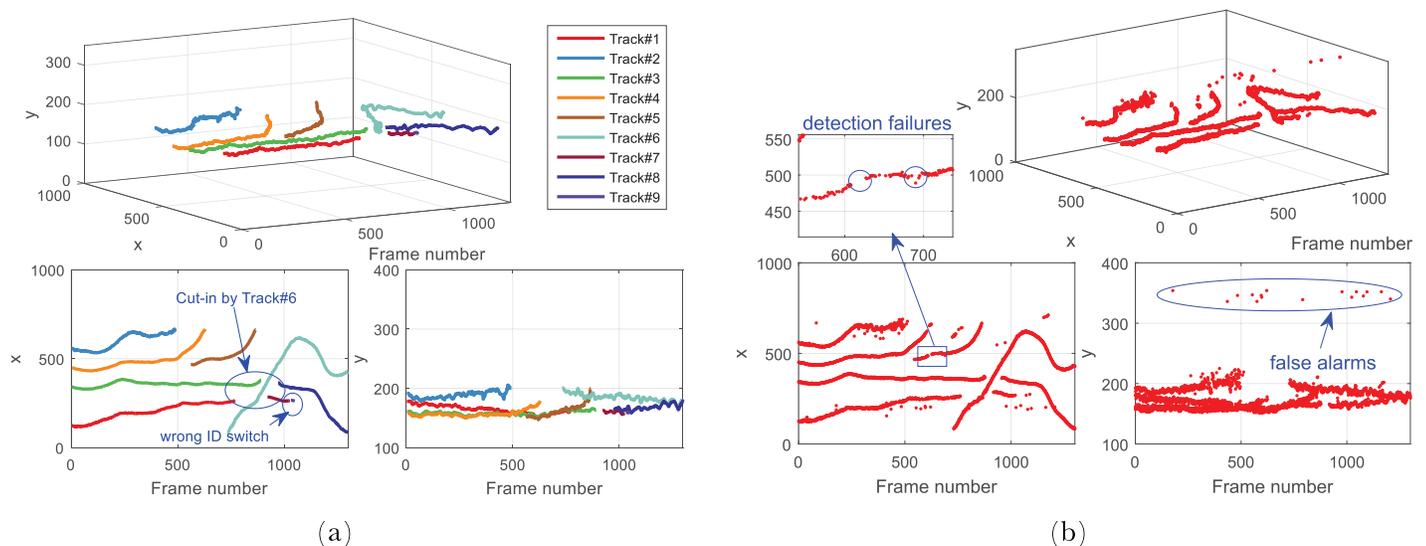


Fig 13. The trajectories of different vehicles in 3-Ring road. (a)The tracking result of our method, and (b)the result of original Cascade Adaboost detector. Note that our method could track each vehicle with correct ID, and efficiently suppress the false positive detections and temporal loss of detection.

<https://doi.org/10.1371/journal.pone.0173424.g013>

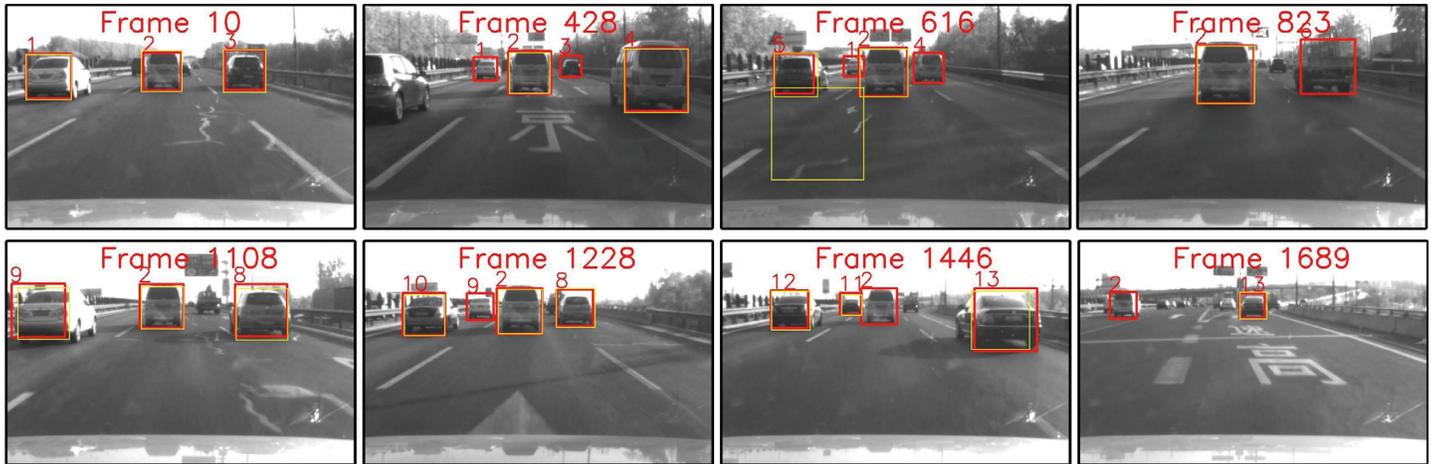


Fig 14. Typical tracking results of the video captured on 5-Ring road. (Yellow) Original cascade Adaboost detector. (Red) Ours. Note that different vehicles are tracked with independent ID numbers throughout the video by our method. Note the detection failures and false alarms of original detector.

<https://doi.org/10.1371/journal.pone.0173424.g014>

The quantitative results of these two videos are shown in Table 3. Compared with original vehicle detector, our method could achieve a better accuracy with lower processing time. It proves that the proposed data association method could correctly assign measurements with existing tracks in every epoch, the track management method could successfully suppress the sparse false alarms(FPR) and pick up the true detections through temporal context. Furthermore, once a track is created, the track management could efficiently remedy the wrong terminations caused by temporary detection failures. It is worth to mention that most of the remaining detection failure(1.3% FNR) happened during the creation of each track(see Frame 964 in Fig 12 for example) since we only took the registered vehicles for evaluation. Actually

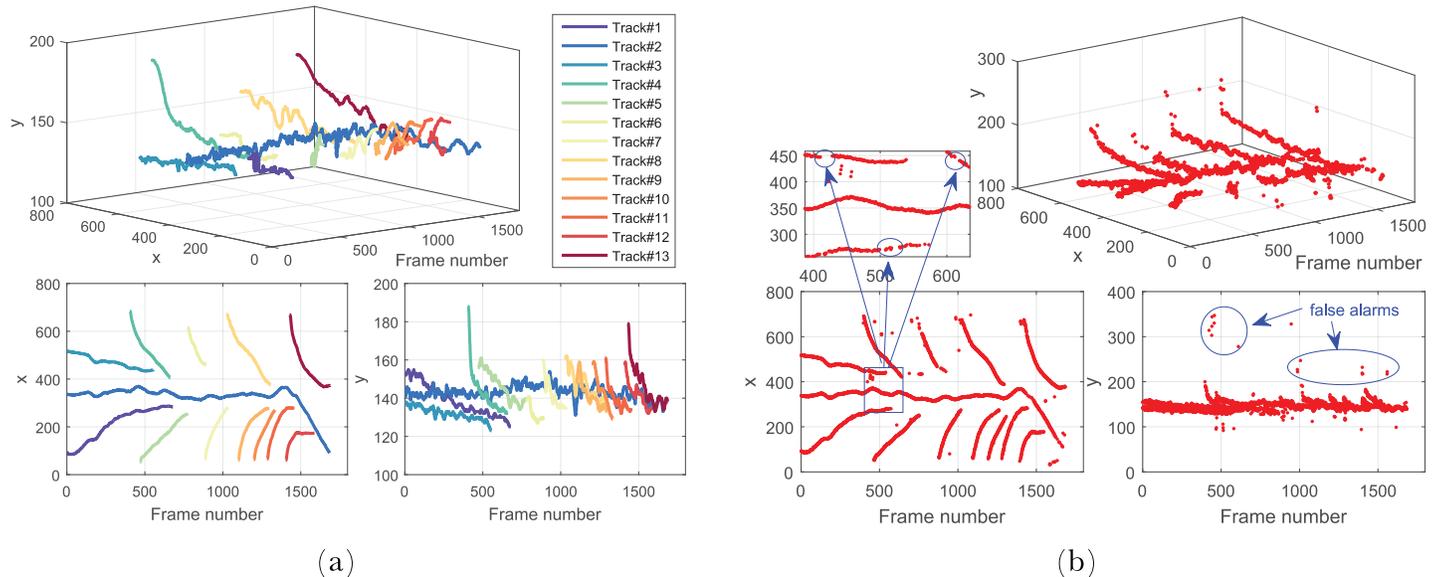


Fig 15. The trajectories of different vehicles. (a)The trajectories of different vehicles produced by our method. (b)The detection result generated by original Cascade Adaboost detector. Our method could track each vehicle with correct ID, and efficiently suppress the false positive detections and temporal loss of detection.

<https://doi.org/10.1371/journal.pone.0173424.g015>

Table 3. Multi-vehicle tracking performance.

Scenarios		FPR	FNR	FPS
3-Ring Road	Original Detector	5.2%	3.4%	4.3
	ProposedMethod	0.1%	1.4%	23.1
5-Ring Road	Original Detector	3.1%	8.9%	4.7
	ProposedMethod	0.2%	1.1%	22.7
Average	Original Detector	4.6%	6.5%	4.5
	ProposedMethod	0.2%	1.3%	22.9

<https://doi.org/10.1371/journal.pone.0173424.t003>

this kind of detection failures had already been successfully detected and observed by the track management. Additionally, the ROI based detection with periodical global searching scheme could reduce the computation costs while find the new appearing targets in time, therefore the frame rate is improved remarkably without losing the detection performance compared with the global searching based original detector.

Limitations of the proposed method

This section discuss the limitations of the proposed method. As shown in Fig 16(a), the proposed method can only detect the fully appeared or slightly occluded vehicles, when the vehicle targets are highly occluded or partially appeared, it could not guarantee a reliable detection. The proposed detector will also lose the detection when there is high contrast background occurred due to sudden illumination changes, as shown in Fig 16(b). Although the tracking mechanism could mitigate the detection loss to a certain degree, the wrong termination of existing tracks are still usually unavoidable unless the camera’s exposure regains balance very quickly. Future work will focus on improving the detection of these more challenging scenarios. Additionally, the proposed data association and track management method utilized many heuristic fixed thresholds for determining the creation and termination of tracks. Future work will also include analysing the sensitivity and influence of these parameters and finding the adaptive threshold for some important parameters.

Conclusion

This paper has proposed an improved multi-vehicle tracking method using cascade Adaboost and adaptive Kalman filter with target identity awareness for driver assistant systems. For vehicle detection, we built a cascade Adaboost classifier using Haar-like features, followed by a



Fig 16. Detection failures. (a) The vehicle in the right lane is not detected due to partial appearance. (b) The vehicle in the ego lane is not detected due to high contrast background.

<https://doi.org/10.1371/journal.pone.0173424.g016>

verification process which could refine the vehicle hypothesis more systematically. In vehicle tracking, the detections of every epoch were assigned by a GNN based association method while considering their local validities, and an adaptive Kalman filter with on-line stochastic modelling was developed for robust state estimation. Additionally, a temporal context based track management was also proposed to suppress spare false detection and avoid wrong termination of existing tracks caused by temporary detection failures. The proposed method was test on various challenging real roads, and the experimental results showed that the vehicle detection performance was greatly improved with lower false alarm rate, lower detection failure rate and higher processing speed. Finally, the limitation of the proposed method and future work were also discussed.

Supporting information

S1 File. The full detection result of 3-Ring Road in video format. (Yellow) Original cascade Adaboost detector, (Red) Ours.
(MP4)

S2 File. The full detection result of 5-Ring Road in video format. (Yellow) Original cascade Adaboost detector, (Red) Ours.
(MP4)

Acknowledgments

This research is sponsored by the National Natural Science Foundation of China under Grant No.51005019.

Author Contributions

Conceptualization: BW ZQ.

Data curation: BW ZQ SC ZL GM.

Formal analysis: BW ZQ.

Funding acquisition: ZQ.

Investigation: BW ZQ SC ZL GM.

Methodology: BW ZQ.

Project administration: BW ZQ.

Resources: BW ZQ.

Software: BW ZQ.

Supervision: SC ZL.

Validation: BW ZQ.

Visualization: BW.

Writing – original draft: BW.

Writing – review & editing: BW ZQ SC ZL GM.

References

1. Sivaraman, S and Trivedi, MM. A review of recent developments in vision-based vehicle detection. *Vehicles Symposium. Citeseer*, 2013, p. 310–5.
2. Matthews N, An P, Charnley D and Harris C. Vehicle detection and recognition in greyscale imagery. *Control Engineering Practice*. 1996; 4: 473–9. [https://doi.org/10.1016/0967-0661\(96\)00028-7](https://doi.org/10.1016/0967-0661(96)00028-7)
3. Tzomakas C and von Seelen W. Vehicle detection in traffic scenes using shadows. IR-INI, INSTITUT FÜR NUEROINFORMATIK, RUHR-UNIVERSITÄT. Citeseer, 1998.
4. Teoh SS and Bräunl T. Symmetry-based monocular vehicle detection system. *Machine Vision and Applications*. 2012; 23: 831–42. <https://doi.org/10.1007/s00138-011-0355-7>
5. Hawe S. A component-based image processing framework for automotive vision. *Master's thesis, University of Western Australia*. 2008.
6. Kourou K, Exarchos TP, Exarchos KP, Karamouzis MV, Fotiadis DI. Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*. 2015; 13:8–17. <https://doi.org/10.1016/j.csbj.2014.11.005> PMID: 25750696
7. Hu X, Li SE, Yang Y. Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles. *IEEE Transactions on Transportation Electrification*. 2016; 2(2):140–9. <https://doi.org/10.1109/TTE.2015.2512237>
8. Graves A, Mohamed A-r, Hinton G, editors. Speech recognition with deep recurrent neural networks. *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on*; 2013: IEEE.
9. Viola P and Jones M. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, 2001 CVPR 2001 Proceedings of the 2001 IEEE Computer Society Conference on*. IEEE, 2001, p. I-511-I-8 vol. 1.
10. Sivaraman S and Trivedi MM. Active learning based robust monocular vehicle detection for on-road safety systems. *Intelligent Vehicles Symposium, 2009 IEEE*. IEEE, 2009, p. 399–404.
11. Stanculescu B, Breheret A and Moutarde F. Introducing new adaboost features for real-time vehicle detection. *arXiv preprint arXiv:09101293*. 2009.
12. Kim G and Cho J-S. Vision-based vehicle detection and inter-vehicle distance estimation for driver alarm system. *Optical review*. 2012; 19: 388–93. <https://doi.org/10.1007/s10043-012-0063-1>
13. Sivaraman S and Trivedi MM. Integrated lane and vehicle detection, localization, and tracking: A synergistic approach. *IEEE Transactions on Intelligent Transportation Systems*. 2013; 14: 906–17. <https://doi.org/10.1109/TITS.2013.2246835>
14. Karasalo M and Hu X. An optimization approach to adaptive Kalman filtering. *Automatica*. 2011; 47: 1785–93. <https://doi.org/10.1016/j.automatica.2011.04.004>
15. Hu X, Sun F, Zou Y. Comparison between two model-based algorithms for Li-ion battery SOC estimation in electric vehicles. *Simulation Modelling Practice and Theory*. 2013; 34:1–11. <https://doi.org/10.1016/j.simpat.2013.01.001>
16. Hu C, Chen W, Chen Y and Liu D. Adaptive Kalman filtering for vehicle navigation. *Positioning*. 2009; 1: 0.
17. Felzenszwalb PF, Girshick RB, McAllester D and Ramanan D. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*. 2010; 32: 1627–45. <https://doi.org/10.1109/TPAMI.2009.167> PMID: 20634557
18. Tuohy S, O'Cualain D, Jones, E and Glavin, M. Distance determination for an automobile environment using inverse perspective mapping in OpenCV. *Proc Irish Signals and Systems Conference*. 2010.
19. Bar-Shalom Y. Tracking and data association. Academic Press Professional, Inc., 1987.
20. Musicki D and Evans R. Joint integrated probabilistic data association: JIPDA. *IEEE transactions on Aerospace and Electronic Systems*. 2004; 40: 1093–9. <https://doi.org/10.1109/TAES.2004.1337482>
21. Reid D. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*. 1979; 24: 843–54. <https://doi.org/10.1109/TAC.1979.1102177>
22. Konstantinova P, Udvarov A and Semerdjiev T. A study of a target tracking algorithm using global nearest neighbor approach. *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'03)*. 2003, p. 290–5.
23. Schubert R, Adam C, Richter E, Bauer S, Lietz H and Wanielik G. Generalized probabilistic data association for vehicle tracking under clutter. *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, p. 962–8.
24. Dehghan A, Modiri Assari S and Shah M. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, p. 4091–9.