

Vision-based Frontal Vehicle Detection and Tracking

King Hann LIM, Kah Phooi SENG, Li-Minn ANG and Siew Wen CHIN

*School of Electrical and Electronic Engineering
The University of Nottingham Malaysia campus, Jalan Broga, Semenyih, 43500, Selangor
Tel : 03-89248350, Fax : 03-89248071
E-mail : {keyx7khl,Jasmine.Seng, Kenneth.Ang, keyx8csw}@nottingham.edu.my*

ABSTRACT

This paper presents a vision-based driver assistance system composing of vehicle detection using knowledge-based method and vehicle tracking using Kalman filtering. First, a preceding vehicle is localized by a proposed detection scheme, consisting of shadow detection and brake lights detection. Second, the possible vehicle region is extracted for verification. Symmetry analysis includes contour and brake lights symmetries are performed and followed by an asymmetry contour analysis in order to obtain vehicle's center. The center of vehicle is tracked continuously using Kalman filtering within a predicted sub-window in consecutive frames. It reduces the scanning process and maximizes the computational speed of vehicle detection. Simulation results demonstrate good performance of the proposed system.

Keywords

Vehicle Detection, Vehicle Tracking, Intelligent Vehicle and Driver Assistance System

1.0 INTRODUCTION

Vehicle detection and tracking approach has become increasingly important as a driver assistance system, leading towards the advancement of autonomous vehicular technology. The detection of preceding vehicles in sudden break, slow-down, and uniform driving speed has to be responded immediately from a driver to prevent any casualties. Due to the reducing cost of optical sensors and increasing speed of microprocessor, vision-based vehicle detection may be a feasible solution to improve the security of vehicle driving and to increase efficient use of driving space. Additionally, the assistance of vehicle tracking may further reduce the processing time of vehicle detection and less interruption of environmental noise. Therefore, a fast and effective vehicle scanning technique is highly required to automatically position a moving vehicle in the frontal view of traffic scene for the sake of realizing the autonomous vehicle driving.

Various vehicle detection and tracking techniques were revealed in for a past decade. proposed a vehicle detection

scheme based on voting approach for the global contour symmetry axis and noise filtering. It was followed by vehicle segmentation to extract a leading vehicle with a bounding box and continuously track the vehicle with local symmetry axis detection. On the other hand, detected vehicle in a traffic scene regarding to a high vertical symmetry using multi-resolution method. They computed symmetries on binarized Sobel modules, almost-vertical edges and almost-horizontal edges (. At the same time, localized preceding vehicles by constructing a geometric model and its energy function about the information of vehicle's shape and symmetry. However, these vehicle detection techniques depending on solely symmetry detection might not generate good detection results and it was time-consuming to search the entire image for symmetry axis detection.

Besides, proposed a preceding vehicle tracking and detection system including road area finding, vehicle footprint extraction and vehicle bounding box extraction. implemented a multi sensor fusion approach for vehicle detection. Shadow and symmetry detection were performed and these information were combined for vehicle tracking. proposed a preceding vehicle detection method based on multi-characteristics fusion by first identifying the preceding vehicles using the bottom shadows and then verifying the region of interest through the fusion of vehicle margin characteristics, texture characteristics and vertical edge symmetry. Finally, vehicle location can be acquired by means of vehicle margin information. Nonetheless, shadow segmentation given in these papers provided poor segmentation results. Moreover, recognized the preceding vehicle on highways using edges correlation method. It needed the difference between the current image and few previous frames in the past and hence, computation of the correlation image was a time-consuming task. presented an integrated framework of on-road vehicle detection through knowledge fusion such as appearance, geometry and motion information over image sequences. It needed prior training process of vehicle appearance on the system. Furthermore, detected vehicles based on the gradient based methods and Adaboost classification. Off-line learning process was required for the vehicle classification. performed an on-road vehicle motion analysis on a traffic scene, including two steps, i.e. incoming vehicle detection and vehicle motion analysis. It would be false detection if the preceding vehicle remained constant speed or halt.

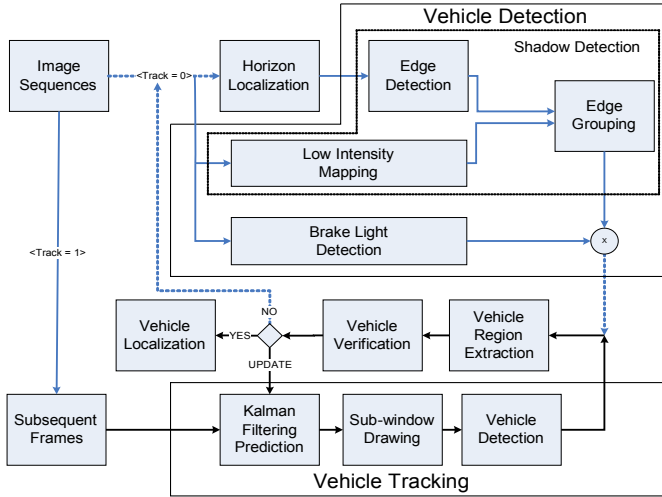


Figure 1: The proposed vehicle detection and tracking system.

In this paper, a preceding vehicle detection and tracking system is proposed in Figure 1. The proposed system composes of three main modules: (i) proposed vehicle detection based on shadow and brake lights detection, (ii) proposed vehicle verification, and (iii) vehicle tracking using Kalman filtering. In the vehicle detection module, a frontal vehicle is detected by first localizing the shadow edge and verifying with red components which usually refers to brake lights. The possible regions are then extracted with a bounding box for vehicle verification. During the verification process, symmetry axes for contour and brake lights are first detected and it is followed by asymmetry contour analysis. Eventually, the center parameter of a vehicle is updated using Kalman filtering in order to predict a sub-window, which limits the scanning process of the vehicle detection. Once the tracking mode is turned on, our proposed system undergoes closed-loop vehicle detection. It reduces the computational time of vehicle detection. If the vehicle is miss-tracked, vehicle detection will then be reprocessed on the entire image.

2.0 VEHICLE DETECTION

Vehicle detection plays a critical role to automatically localize a vehicle in a preceding view of image. Its purposes are to improve the vehicle safety and stay a certain distance away from a frontal vehicle avoiding severe crash. In the proposed vehicle detection method, it can be divided into three stages, i.e. horizon localization, shadow detection, and brake lights detection.

2.1 Horizon Localization

A horizon line is localized using vertical mean distribution to split a traffic scene image $I(x,y)$ into sky region and road region where x and y depict the row and column for the

original image. Vertical mean distribution, showing in Figure 2(b) is measured by averaging the gray values of each row on the $I(x,y)$. The threshold value for horizon line is obtained through a minimum search along the vertical mean curve, where the first minimum occurs from the upper curve is the regional dividing line. A road image (R_{roi}) is generated in Figure 2(a) after the application of horizon line threshold, where all vertical coordinates below the threshold are discarded. This stage is only applicable temporary if the condition $\langle \text{Track} = 0 \rangle$ is true.

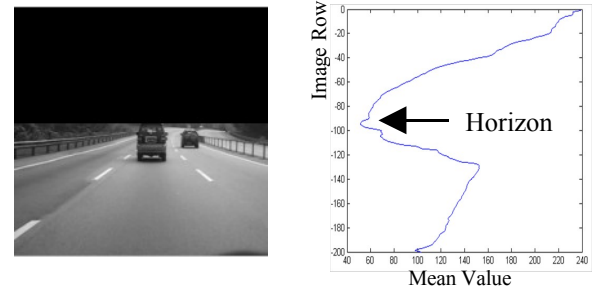


Figure 2: (a) R_{roi} , (b) Vertical mean distribution.

2.2 Shadow Detection

Shadow is usually found underneath a vehicle in an image. In this stage, a shadow detection method is proposed to quickly scan for the low intensity values existed in the image. The proposed shadow detection method has three steps, i.e. edge detection, low intensity mapping and edge grouping.

2.2.1 Edge Detection

First, a mean filter is applied to the R_{roi} map to smooth image pixel values and enlarge shadow segments. Next, edge detection with a Sobel filter is performed on the blurring image to obtain the contour of a vehicle as well as its shadow. Subsequently, a certain edge threshold (T_l) is applied to the gradient edge image to produce a binarized edge map $BEM(x,y)$, as demonstrated in Figure 3.

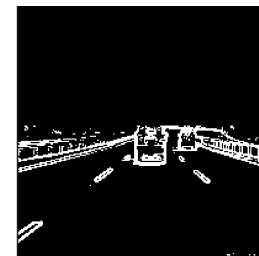


Figure 3: Binarized edge map (BEM).

2.2.2 Low Intensity Mapping

Shadow is always existed in an image with low intensity values. To produce a low intensity map $LIM(x,y)$

automatically, the mean value (μ) and standard deviation (σ) of $I(x,y)$ is initially computed and it is then determined as shown in Figure 4(a) with the equation (1),

$$LIM(x,y) = \begin{cases} 1 & ,if I(x,y) - \mu < -\sigma \\ 0 & ,otherwise \end{cases} \quad (1)$$

Subsequently, the possible shadow map $PSM(x,y)$ illustrated in Figure 4(b) is obtained by merging the $BEM(x,y)$ and $LIM(x,y)$ with an AND operation.

$$PSM(x,y) = \begin{cases} 1 & ,if LIM(x,y) == 1 AND BEM(x,y) == 1 \\ 0 & ,otherwise \end{cases} \quad (2)$$



Figure 4: (a) LIM(x,y), (b) PSM(x,y).

2.2.3 Edge Grouping

After the acquisition of $PSM(x,y)$, edge grouping stage is performed to acquire the possible shadow edge map $PSEM(x,y)$. First, a noise elimination technique is applied to group non-zero connectivity edges with a pixel count threshold (T_2) and at the same time, discard the non-connectivity pixels that having less than T_2 pixels count. To generate $PSEM(x,y)$, only the bottom edge of the connected pixels are remained since shadows are always existed underneath a vehicle. Hence, the remaining shadow edges are labeled accordingly and it is shown in Figure 5.

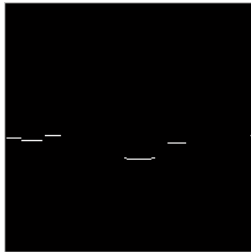


Figure 5: Possible shadow edge map (PSEM).

2.3 Brake Lights Detection

Brake lights have much red components compared to other vehicle's parts. Figure 6(a) depicts the result of red component map. $PSEM(x,y)$ is checked by accumulate the number of red component pixels for each labeled edge. If the edge has more than T_3 number of red component pixels, it is defined as a possible vehicle region (PVR) as shown in Figure 6(b).



Figure 6: (a) Red component map, (b) Possible vehicle region (PVR).

3.0 VEHICLE VERIFICATION

In order to detect a preceding vehicle, only the largest area of PVR is concerned and verified for the tracking system. A bounding box is applied to the possible vehicle region based on the edge width obtained from $PSEM(x,y)$. Figure 7 shows the PVR extraction with the largest area. The extraction region is then undergone symmetry and asymmetry analysis for PVR verification.



Figure 7: PVR extraction.

3.1 Symmetry Analysis

Two symmetry analysis are performed, i.e. contour symmetry and brake lights symmetry detection. Assuming that the extraction region has i th row and j th column of image size. A binary contour (C_{PVR}) is first constructed by using Sobel filter as shown in Figure 8(a). For the non-zero pixels (i, j_1), the contour symmetry axis is calculated respective to the consecutive pixel (i, j_2) at the same row .

$$center = \frac{j_1 + j_2}{2} \left. \vphantom{\frac{j_1 + j_2}{2}} \right\} \text{if } C_{PVR}(i, j_2) == 1 \text{ AND } j_1 \neq j_2 \quad (3)$$

$$C_{sym} = \arg \max_j ContourSym(j) \quad (4)$$

where $ContourSym$ is an accumulator and C_{sym} is the contour symmetry axis, in which the maximum peak occurs in Figure 8(b). On the other hand, brake lights are always located on the left and right corner of a vehicle symmetrically. Figure 8(c) is the detected binary brake lights (B_{PVR}) on the extraction region. Same symmetry voting scheme is applied to the brake lights symmetry analysis with an accumulator called $BrakeSym$.

The brake lights symmetry axis is defined as:

$$B_{sym} = \arg \max_j BrakeSym(j) \quad (5)$$

where B_{sym} is the maximum peak obtained in the accumulator demonstrated in Figure 8(d). To verify the validity of PVR , the condition below must be obey:

$$Track = \begin{cases} 1, & \text{if } |B_{sym} - C_{sym}| \leq \varepsilon \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where ε is the difference between brake lights symmetry axis and contour symmetry axis.

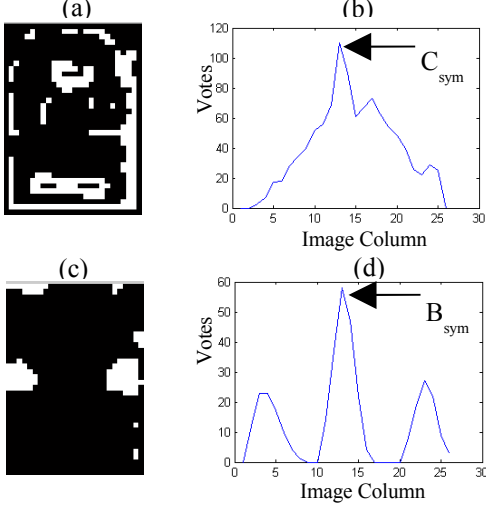


Figure 8: (a) Binary Contour (C_{PVR}), (b) ContourSym, (c) Binary Brake lights (B_{PVR}), (d) BrakeSym.

3.2 Asymmetry Analysis

As can be seen in Figure 8(a), the edges of vehicle region are mostly displayed in vertical, but not in horizontal. Instead of analyzing the grayscale symmetry, the vehicle region is turned into an edge difference map by differencing all the columns to its first column of the PVR where it is shown in Figure 9(a). Next, a Sobel filter is applied to the difference map to get the horizontal contour (D_{PVR}), as shown in Figure 9(b). For the non-zero pixels (i_1, j) , the contour symmetry axis is calculated regarding to the consecutive pixel (i_2, j) at the same column as below:

$$D_{PVR}(i, j) = \nabla |G(i, j) - G(i, 1)| \quad (7)$$

$$center = \frac{i_1 + i_2}{2} \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{if } D_{PVR}(i_2, j) = 1 \text{ AND } i_1 \neq i_2 \quad (8)$$

$$D_{asym} = \arg \max_i ContourAsym(i) \quad (9)$$

where $ContourAsym$ is an accumulator and D_{asym} is the contour asymmetry axis. By combining the symmetry and asymmetry

axis, we might find the center coordination as (C_{sym}, D_{asym}) of the PVR .

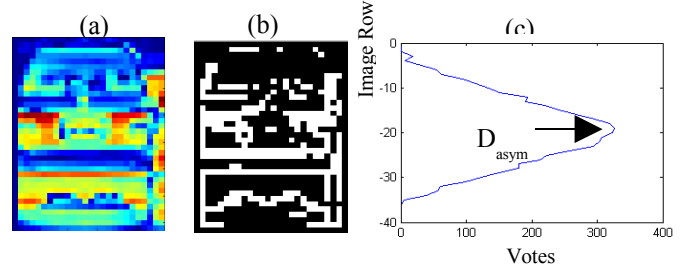


Figure 9: (a) Edge difference map, (b) D_{PVR} , (c) ContourAsym.

4.0 VEHICLE TRACKING

Vehicle tracking is supplementary to support a driver assistance system for fast vehicle detection. As shown in Figure 10, vehicle tracking algorithm is implemented in a closed loop if $\langle Track = 1 \rangle$ is true. Once the system lost track, vehicle detection will be carried out on the whole image.

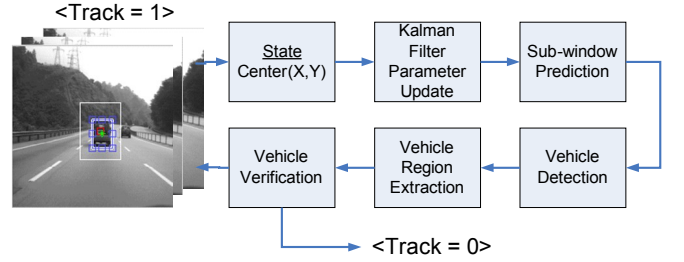


Figure 10: Vehicle tracking system.

Kalman filter is popular in vision tracking system due to its prediction characteristic based on previous information. The linear state equation and measurement equations are defined as :

$$x_{k+1} = A_{k+1,k} x_k + B_{k+1,k} u_k + w_k \quad (10)$$

$$y_k = H_k x_k + v_k \quad (11)$$

where the state space (x) contains the X-position and Y-position of vehicle's center; $A_{k+1,k}$ is the transition matrix bringing state x_k from time k to $k+1$, $B_{k+1,k}$ is the matrix connecting the acceleration vector u_k to the state variables; w_k is known as process noise; y_k is the measurement output; H_k is the observation model that maps the true state space to the observed space; and v_k is the measurement noise. The initial conditions for \hat{x}_k and P_k are given as:

$$\hat{x}_0 = E[x_0] \quad (12)$$

$$P_0 = E[(x_0 - E[x_0])(x_0 - E[x_0])^T] \quad (13)$$

In Figure 11, P_k^- is a priori estimate error covariance and P_k is a posteriori estimate error covariance. The estimated state, x_k is

updated with the estimated gain G_k . The predicted state x_k^- is used to draw a sub-window on the next frame.

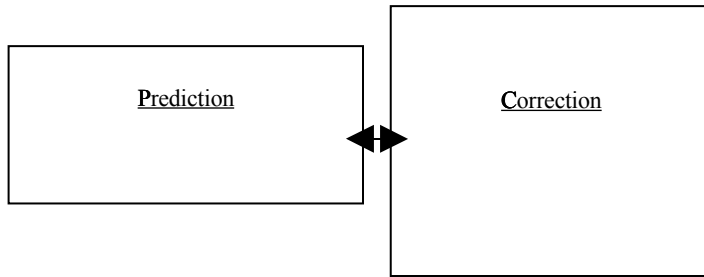


Figure 11: Kalman filtering method.

5.0 RESULTS AND DISCUSSIONS

In this section, some results were presented to evaluate the performance of the proposed algorithm. Two video sequences were captured under sunny day condition at around 12p.m. using 6 Megapixels Canon IXUS 65. Video #1 was taken in a highway environment while video #2 was taken in a rural area. Initial values of the following parameters, $T_1 = 50$, and $T_2 = T_3 = 12$ were set in these experiments.

The estimation of X-position and Y-position for a vehicle's center was demonstrated in Figure 12. For some brief explanation, desired point was the expected center on the vehicle region while the actual point was the position determined from the vehicle verification. Moreover, the predicted point was the position forecasted using the Kalman filtering. As can be observed in Figure 12(a) for video #1, the actual X-position and Y-position were detected approximately along with the desired one within tolerance ± 5 pixels. Same performance was obtained in Figure 12(b) for video #2 with the proposed vehicle verification techniques.

As shown in Figure 13 & 14, it showed a good performance of the proposed vehicle detection and tracking system since a preceding vehicle was well detected and tracked

consecutively. The center of vehicle was well detected with a "*" symbol while "□" denoted the predicted center tracked with Kalman filter. In Table 1, the detection time taken in assistance with vehicle tracking was (~90%) much faster than only vehicle detection in the traffic scene. The recommended safe distance is the summation of mean tracking time distance [$\sim 0.12 * speed$] and vehicle stopping distance [$speed^2/20$] regarding to the current speed as reported in (June, 2000).

Table 1: Time taken for vehicle detection and tracking

Video	Total Frames	Detection Time (s)	Detection+ Tracking Time (s)	Mean Time(s)
#1	170	1.24~1.82	0.11~0.13	0.12
#2	190	1.25~1.82	0.14~0.16	0.15

6.0 CONCLUSION

A vision-based preceding vehicle detection and tracking system has been presented in this paper. Frontal vehicle is first detected based on shadow and brake lights detection. A bounding box extraction is applied to the possible vehicle region for symmetry verification. Brake lights symmetry and contour symmetry analysis are implemented and followed by contour asymmetry analysis in order to obtain center of a possible vehicle region. It is then passed to the tracking system to predict the movement of a vehicle. The advantages of this vehicle detection and tracking system are, (i) the reduction of vehicle searching time, and (ii) the increase performance of the vehicle detection since the searching area is bounded. However, this method cannot be applied during night or poor lighting condition and far vehicle detection.

REFERENCES

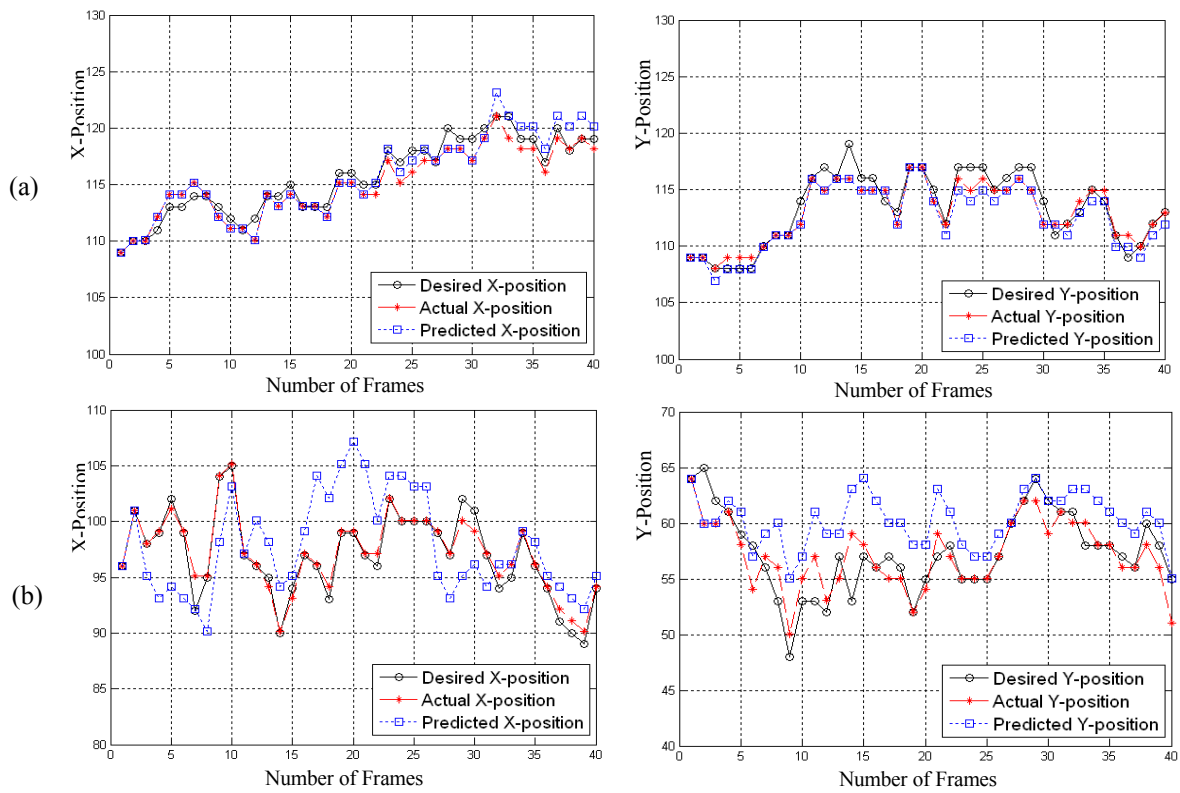


Figure 12: The estimation of X-position and Y-position for (a) Video #1, (b) Video #2.

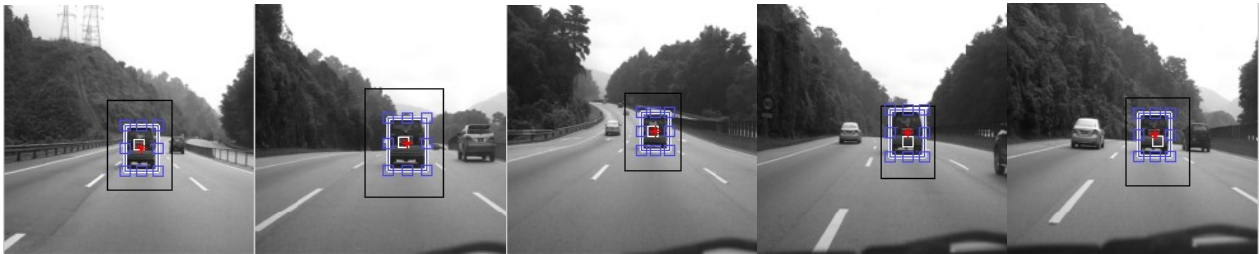


Figure 13: Results of vehicle detection and tracking for video #1
(a) Frame 5, (b) Frame 50, (c) Frame 100, (d) Frame 140, (e) Frame 170

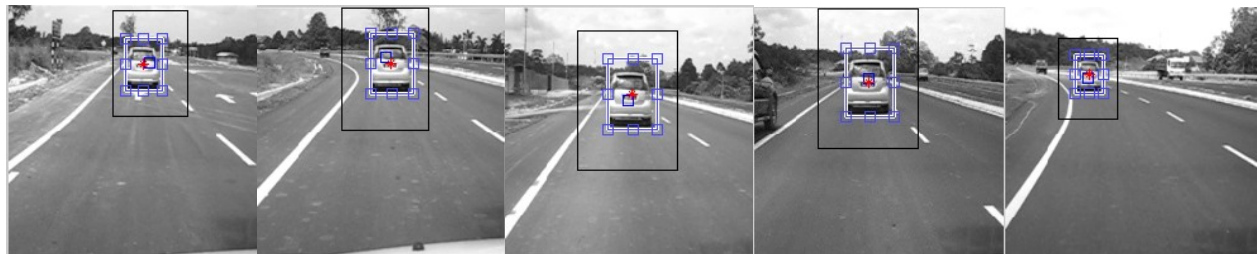


Figure 14: Results of vehicle detection and tracking for video #2
(a) Frame 5, (b) Frame 50, (c) Frame 100, (d) Frame 140, (e) Frame 190