

## Research Article

# A Real-Time Embedded Blind Spot Safety Assistance System

**Bing-Fei Wu, Chih-Chung Kao, Ying-Feng Li, and Min-Yu Tsai**

*Institute of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan*

Correspondence should be addressed to Chih-Chung Kao, kevinkao@cssp.cn.nctu.edu.tw

Received 14 October 2011; Accepted 9 January 2012

Academic Editor: David Fernández Llorca

Copyright © 2012 Bing-Fei Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an effective vehicle and motorcycle detection system in the blind spot area in the daytime and nighttime scenes. The proposed method identifies vehicle and motorcycle by detecting the shadow and the edge features in the daytime, and the vehicle and motorcycle could be detected through locating the headlights at nighttime. First, shadow segmentation is performed to briefly locate the position of the vehicle. Then, the vertical and horizontal edges are utilized to verify the existence of the vehicle. After that, tracking procedure is operated to track the same vehicle in the consecutive frames. Finally, the driving behavior is judged by the trajectory. Second, the lamps in the nighttime are extracted based on automatic histogram thresholding, and are verified by spatial and temporal features to against the reflection of the pavement. The proposed real-time vision-based Blind Spot Safety-Assistance System has implemented and evaluated on a TI DM6437 platform to perform the vehicle detection on real highway, expressways, and urban roadways, and works well on sunny, cloudy, and rainy conditions in daytime and night time. Experimental results demonstrate that the proposed vehicle detection approach is effective and feasible in various environments.

## 1. Introduction

In recent years, the driving safety has become the most important issue in Taiwan. Numbers of car accidents and casualties increase year by year. According to the accident data by Taiwan Area National Freeway Bureau, the main reason of occurring accidents is human negligence. Therefore, collision-forewarning technologies get great attention increasingly, and several kinds of driving safety assisted products are promoted, including lane departure warning systems (LDWSs), blind spot information systems (BLISs), and so forth. These products could provide more information about the vehicle surroundings with the driver, so that the driver could make the correct decision when driving on the road. BLIS could monitor whether the vehicles appear in the side of host car or not and inform the driver when the driver intends to change lanes. Radar is another solution for BLIS. However, the cost is much higher than the camera. Consequently, vision-based blind spot detection becomes popular in this field.

There are many vision-based obstacle detection systems proposed in the literature. Most of them focus on detecting lane marking, the obstacles in front of the host car for lane

departure warning [1–3], and collision avoidance applications [4–7]. Lane detection is exploited for driving safety assistant in early periods. Moreover, a complete survey was addressed in [1]. Besides, the front obstacle detection was discussed enthusiastically in the past decade. Online boosting algorithm is proposed to detect the vehicle in front of the host car [2]. The online learning algorithm can conquer the online tuning problem for a practical system. O'Malley et al. [3] presented a rear-lamp vehicle detection and tracking for night condition. The rear-lamp pairs are used to recognize the front vehicle and track lamp pairs by Kalman filter. Liu and Fujimura [4] proposed a pedestrian detection system by stereo night vision. Human became hot spot in night vision and would be tracked by blob matching. Labayrade et al. [5] integrated 3D cameras and laser scanner to detect multiobstacles in front of the vehicle. The width, height, and depth of the obstacle were estimated by stereo vision and the precise obstacle position can be provided by laser scanner. This cooperative fusion approach achieved an accurate and robust detection.

Although most attention was fascinated with the front-view obstacle and lane detection, some researchers attended to solve the problems in blind spot obstacle detection.

Wong and Qidwai [6] installed six ultrasonic sensors and three image sensors in the car. They applied fuzzy inference in algorithm to forewarn the driver and reduce the car accident possibility. Achler and Trivedi [7] mounted an omnidirectional camera to monitor the area surrounding by the host vehicle. Hence, the obstacles in both sides of blind spot could be detected at the same time. The wheel was filtered with Type Filter and then the system could determine whether the vehicle exists or not. Ruder et al. [8] designed a lane change assistance system with far range radar, side radar, and stereo vision. The sensor fusion and Kalman filter were used to track the vehicle stably. Díaz et al. [9] applied optical flow algorithm to segment the vehicle in the blind spot, and several scale templates were established for tracking. Batavia et al. [10] also monitored the vehicle in rear image with optical flow algorithm and edge feature. Stuckman et al. [11] used infrared sensor to get the information of the blind spot area. This method was implemented on digital signal processor (DSP) successfully. Adaptive template matching (AdTM) algorithm [12] was proposed to detect the vehicle which was entering the blind spot area, and the algorithm defined levels to determine the behavior of the tracked vehicle. If this vehicle was approaching, the level would increase, otherwise the level would decrease. Multi-line-CCD was equipped by Yoshioka et al. [13] to monitor the blind spot area. This sensor could obtain the height of a pixel in the image because of the parallax between two lenses. Thus, this method could obtain the height of the vehicle. Techmer [14] utilized inverse perspective mapping (IPM) and edge extraction algorithm to match the pattern and to determine whether a vehicle exists in the blind spot or not. Furukawa et al. [15] applied three cameras to monitor front area, left behind area and right behind area. Horizontal segment by edge was used and the template matching was operated by orientation code matching, which is one of the robust matching techniques. Most importantly, these algorithms required less resources for operation and were implemented in one embedded system. Jeong et al. [16] separated the input image into several segmentations and determined these segmentations belonging to the foreground or background by gray level. Afterwards, scale invariant feature transform (SIFT) was implemented to generate robust features to check whether the vehicle exists or not. Finally, modified mean-shift was used to track the detected vehicle. C. T. Chen and Y. S. Chen [17] estimated the image entropy of the road scene in the near lane. The obstacle could be detected and located by analyzing the lane information. Although they could track the obstacles in real-time, they only judged whether the tracked vehicle was approaching or not by considering the location in the previous frame and current frame. Consequently, the false alarm would be easily triggered. Four prespecified regions were defined to identify the dangerous level in [18]. Sobel edge was extracted and morphological operation was applied to generate clearer edge image. However, if only considering the edge information, the system would easily alarm falsely by shadow and safety island.

The blind spot detection (BSD) for vehicles and motorcycles in daytime and nighttime is proposed in this paper.

Moreover, one of the most important issues in this field is the execution efficiency of the system. If the efficiency is not high enough, this system could not achieve it to real-time results, and it could not remind the driver immediately. There would be no use value in the system which is described above. There are a lot of methods which had been proposed in recent years to prevent collision in blind spot zone, but most of those methods are implemented on PC, which are not suitable as an automobile electronics. Although there are some methods which were implemented on DSP platform, low frame rate and robustness became the serious problems to them. In this paper, edge, shadow, and lamps in spatial domain are applied to increase execution efficiency. Therefore, the performance of vehicle detection in this system is the main topic here, especially overcoming the complex problems in harsh environments, such as driving on urban roads. Using the general features in spatial domain and keeping high performance has been implemented through the method introduced in this paper. Developing on DSP platform, the frame rate of this system could still achieve 59 fps at most. For CIF images, this efficiency is high enough that the system can provide real-time information with the drivers, so that drivers can make the most correct decisions in time. The system has high frame rate on TI DM6437 platform, and through the long verification with on-road field tests on highways, expressways, and urban roadways, and works well on sunny, cloudy, and rainy conditions in the daytime and nighttime. This shows that the system has very well robustness, so that it can work anytime at everywhere and provide warning function with the drivers. The warning functions which could be a buzzer or LED light would be triggered to alarm to the driver.

Section 2 briefly introduces the working flow of the presented method in this paper. The algorithms of vehicle detection in the daytime and nighttime are introduced in Sections 3 and 4, respectively. The experiment results and comparisons would be shown in Section 5. Finally, the conclusions would be addressed in Section 6.

## 2. System Overview

Since the features for vehicle detection in the daytime are obviously different from the features in the nighttime, the utilized features and verification procedures would be distinct. Moreover, considering the practical application, BSD should work day and night. Because it is very difficult to distinguish between the daytime and the nighttime, both of vehicle detection algorithms in the daytime and in the nighttime should be processed in each frame. Daytime and nighttime algorithms which detect and track different features with the same workflow in Figure 1 have been implemented into our system and make this system more practical and robustness. The algorithm for the nighttime follows the algorithm for the daytime in our system, so that there is no need to determine what time it is now.

There are three main detecting modes in the algorithm of vehicle detection. Those are the full searching mode, the tracking mode, and the partial searching mode, respectively.

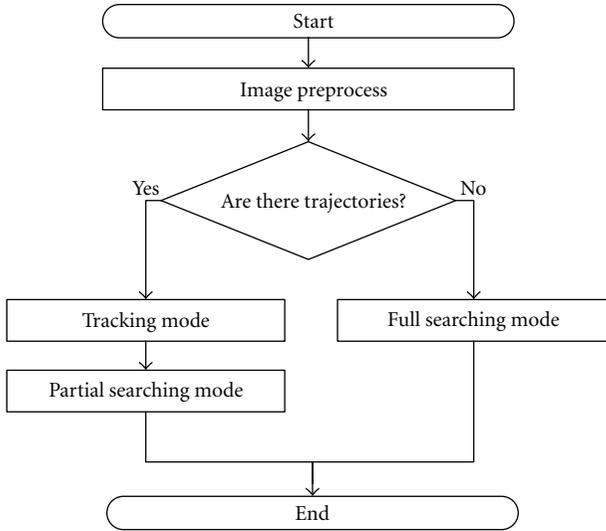


FIGURE 1: The flow chart of the proposed system.

Image preprocessing is performed to extract the edge, shadow, lamp features for the vehicle detection. First, there is no vehicle which is detected and tracked as a trajectory. Therefore, the system would search the possible vehicle candidates in the whole region of the interest (ROI) of the image in full search mode. If the vehicle is detected and tracked successfully in the successive video frames, the vehicle trajectory is generated, and the system would process the tracking mode in the next frame. Because of the data saved from the full searching mode, we already know where the vehicles exist; thus, there is no need to search the whole ROI again. We can only search the region where the vehicles exist and determine their behavior in the tracking mode. According to the locations of vehicles which had been saved in the last frame, the searching region would be set adaptively. After detecting, candidate matching and the vehicle behavior would be judged. In the end, the system triggers the warning signal to remind the driver. Partial searching mode always follows the tracking mode to search if there is any other vehicle or motorcycle in ROI. However, the partial searching mode would not search the zone in where there is already a vehicle appearing and has been detected by the tracking mode.

### 3. Vehicle Detection Algorithm in the Daytime

There are seven topics presented in the following subsections. First, we introduce the definition of our ROI and give comparison with ISO specification in Subsection 3.1. The follows in Subsections 3.2 and 3.3 are edge detection and the shadow searching. After detecting a vehicle, we search the correct boundary of a vehicle in the Subsection 3.4. Then, the candidates are verified in Subsection 3.5. Matching the vehicle appearing in the consecutive frames to generate the vehicle trajectory would be discussed in Subsection 3.6. Finally, the vehicle behavior judgment is performed in the Subsection 3.7.

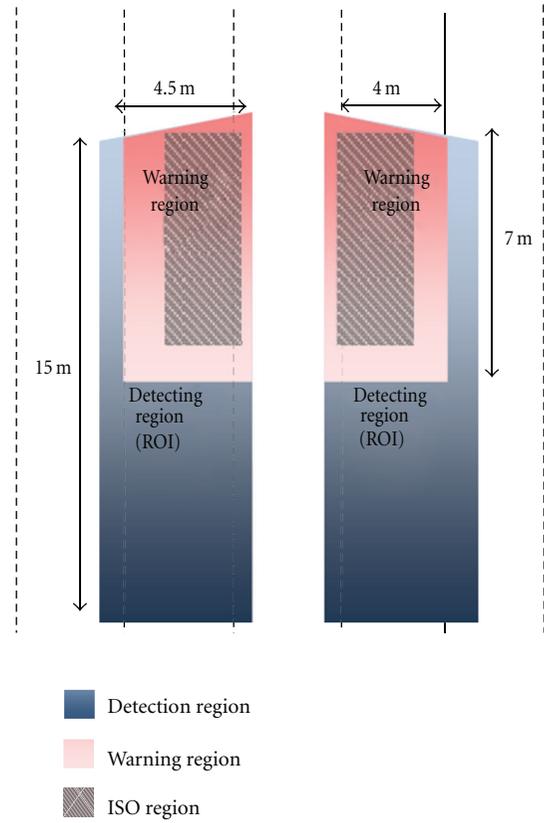


FIGURE 2: Definitions of 3 regions.

3.1. *Define the ROI in the Daytime.* We have to define ROI in this system clearly at first. Referring to the definition of lane change decision aid systems (LCDASS) in ISO document, we obtain the definition of blind spot area and delimit ROI in our system. As shown in Figure 2, the ISO definition of blind spot area is 3 meters to the side of the car and 3 meters behind the car. The ROI in our system is larger than the ISO definition and completely covers it.

The specification of detecting region in this system is 4.5 meters to the side of the car and 15 meters behind the car. The specification of the warning region is 4 meters to side of the car and 7 meters behind of the car. When a vehicle is approaching to the warning region, the system would send a warning signal to the driver. The detecting region and the warning region are drawn in blue and red in Figure 3, respectively.

3.2. *Image Preprocessing for Vehicle Detection in the Daytime.* Shadow and edge features are chosen for vehicle detection in the proposed system in the daytime. Shadow under the car could illustrate the location of the vehicle. Therefore, extracting the shadow region is the first step to detect the vehicle. Wheels always provide great amount of vertical edge, and there are a lot of horizontal edges on the air dam of most vehicles. The information would be fairly useful when detecting vehicles.

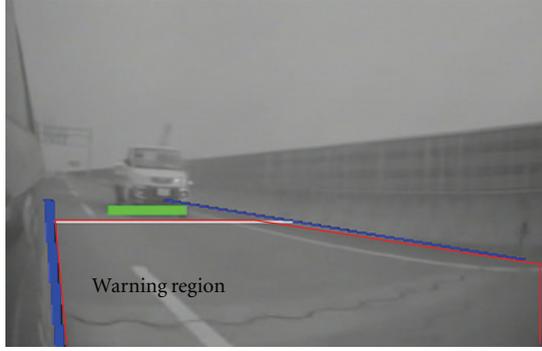


FIGURE 3: The area surrounded by the blue line is detecting region, and the area surrounded by the red line is warning region.

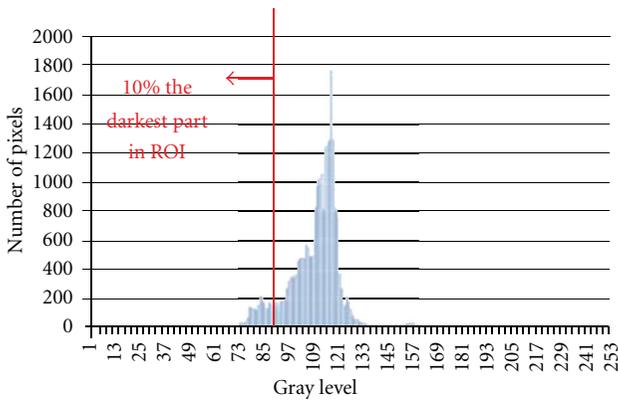


FIGURE 4: The gray level histogram with the pixels in ROI. The darkest part of this histogram might be shadow.

Several weather conditions may happen when driving on the road, so a fixed threshold to extract shadow region may fail in outdoor scene. However, no matter what the weather is, it is supposed that the color of shadow must be the darkest part in ROI. Therefore, we establish a gray level histogram of the pixels in ROI for adaptive shadow threshold. As shown in Figure 4, we assume that 10% of the darkest part of this histogram might be shadow, thus adaptive threshold  $g^*$  for shadow detection in this frame is calculated by (1).  $N$  is the number of total pixels in ROI,  $h(g)$  is the number of pixels at gray level  $g$ , and 0.1 is chosen for  $\delta$  here. According to the states of the road surface, the threshold for shadow extraction could be set dynamically to 92 in the sunny day, 78 in the rainy day, and 56 under the bridge by this method:

$$g^* = \arg \left( \sum_{g=0}^{255} h(g) > \delta N \right). \quad (1)$$

In addition, edge is a useful feature in spatial domain for vehicle detection, and Sobel mask is used to extract the edge feature here. Therefore, there are three kinds of features that could be used to recognize vehicles.

As in Figure 5(a), shadow pixels are drawn in white, and the horizontal edge pixels are drawn in gray. Otherwise, the pixels are set in black. The extracted vertical edge pixels

would be set to 0 in another plane, and the nonvertical edge pixels are set in 255 in Figure 5(b).

**3.3. Shadow Searching.** The first step of vehicle detection is shadow searching. Every pixel between point  $S_L$  and point  $S_R$  in each row is checked. Although every pixel is checked, the computation loading would not increase very much. The definitions of points are calculated by (2) and illustrated in Figure 6, where  $L_P$  and  $R_P$  are the positions of the left and right boundaries of ROI in each row, and  $v$  is the row index:

$$\begin{aligned} S_M(v) &= \frac{(L_P(v) + R_P(v))}{2}, \\ S_L(v) &= \frac{(L_P(v) + S_M(v))}{2}, \\ S_R(v) &= \frac{(R_P(v) + S_M(v))}{2}, \\ S_{LRM}(v) &= \frac{(S_L(v) + S_M(v))}{2}. \end{aligned} \quad (2)$$

When one of these pixels is the shadow pixel, vertical projection would be processed in this row to find whether there are continuous shadow pixels in this row, as shown in Figure 7. The length of shadow pixels and the length of ROI in this row are denoted by  $\lambda_S$  and  $\lambda_L$ , respectively. If  $\lambda_S$  is larger than  $1/8 \lambda_L$ , it would be considered as shadow under a vehicle and keep searching the rest upper part of the ROI. After that, there might be several shadow candidates that have to be confirmed later.

**3.4. Correct Boundaries of the Vehicle.** Although the location of shadow under the vehicle is found, several severe conditions would lead to incorrect detection. Early in the morning or in the evening, the sun would irradiate with a certain angle instead of direct sunlight. This would cause the shadow under a vehicle to become elongate. In the rainy days, severe road surface reflection occurs and causes the same situation as described above, as exhibited in Figure 8. Therefore, the boundaries of vehicle should be confirmed again using average intensity and vertical edges.

In Figure 9, the length of shadow pixels  $\lambda_S$  is found in Subsection 3.3. The row of shadow found in Subsection 3.3 is not the real bottom of the car. Therefore, we have to correct the location from Subsection 3.3. The searching zone is extended  $1/2 \lambda_S$  upward from the row  $RS$ , and the real bottom is searched in this zone  $L'R'RL$ . Shadow should be the darkest intensity because of the shadow property. In this zone, the average gray level value in each row is calculated, and the row with the minimum gray level would be considered as new bottom of the vehicle. The new bottom of the vehicle is obtained in (3), where average gray level value in the  $j$ th row is denoted as  $I(j)$ . Hence, we could update the bottom location of the vehicle to  $v^*$  where is much closer to the bottom of the car:

$$v^* = \operatorname{argmin}_{j=1 \sim (1/2)\lambda_S} \bar{I}(j). \quad (3)$$

After obtaining the bottom position of the vehicle, the left boundary would be searched through the vertical



(a)



(b)

FIGURE 5: (a) The feature plane with horizontal edge and shadow. (b) The vertical edge feature plane.

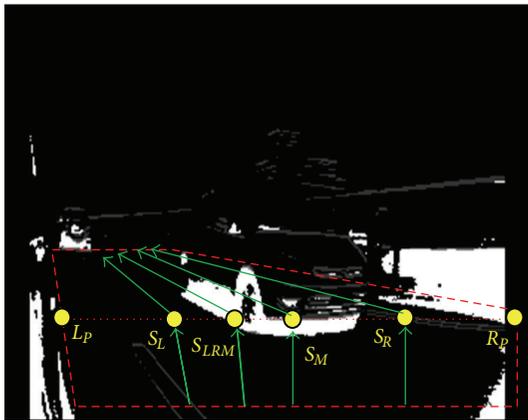


FIGURE 6: The illustration of four shadow searching points.

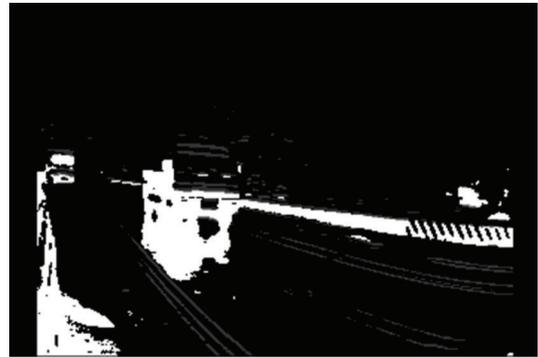


FIGURE 8: The shadow under a vehicle in feature plane is elongated in rainy days.



FIGURE 7: Finding the bottom of the vehicle by vertical projection.

edges. If the continuous vertical edges exist, it would be considered as a vehicle candidate. As seen in Figure 10, the horizontal projection would be performed to check if there are continuous vertical edges of wheels in this region.

The default height of vehicle,  $\lambda_H$ , which is  $3/4 \lambda_S$ , that is, the width of the shadow in Subsection 3.3. The position of the left boundary is computed in (4):

$$u_L = \underset{u=1 \sim \lambda_s}{\operatorname{argmax}} h(u). \quad (4)$$

The amount of vertical edge after horizontal projection is  $h(u)$ . The column of the maximum amount of vertical edge is  $u_L$ . If  $h(u_L)$  is larger than  $1/4 \lambda_H$ , it would be considered as

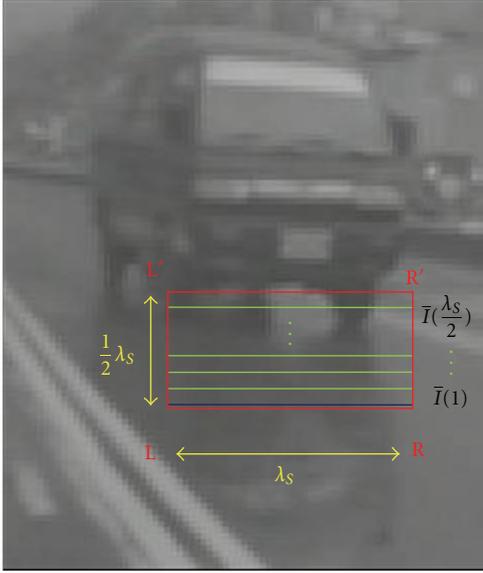


FIGURE 9: The shadow under a vehicle in feature plane is elongated in rainy days.

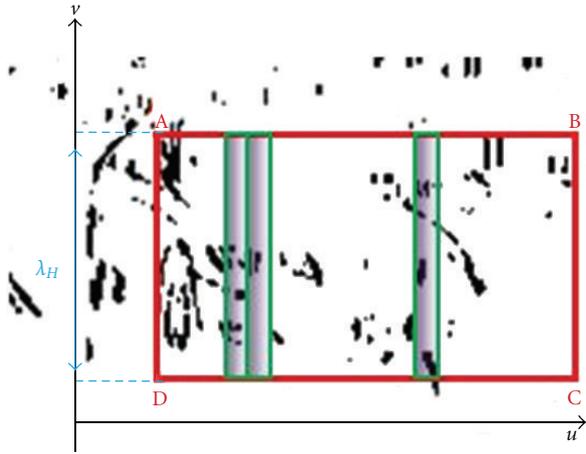


FIGURE 10: The horizontal projection of the vertical edge.

the vertical edge of the wheel. The left side of the boundary of the vehicle would be updated to the column  $u_L$ . The vehicle wheel in right side which is always cloaked from the air dim is hard to find through the edges. Therefore, the right side boundary of the vehicle uses the default value from  $\lambda_s$ . When finish checking up all the shadow candidates with vertical edge, there are some vehicle candidates that have been filtered out here. Next step is confirming all vehicle candidates with horizontal edge characteristic.

**3.5. Vehicle Verification.** Although the shadow under the vehicle and vertical edges is used to detect the vehicle, the verification procedure should be performed to confirm the detection. The horizontal edge is a good feature for

vehicle verification. There are a lot of horizontal edges in the air dam of the most vehicles, so we search the horizontal edge for vehicle verification. As shown in Figure 11, we extend a region for searching continuous horizontal edges. Vertical projection would be processed to check if there are continuous horizontal edges of air dim in this region. The vehicle is verified by (5) and (6):

$$v^* = \operatorname{argmax}_{v=1 \sim \lambda_H} h(v), \quad (5)$$

$$\alpha = \begin{cases} \text{true,} & \text{if } h(v^*) > \frac{\lambda_s}{4}, \\ \text{false,} & \text{otherwise.} \end{cases} \quad (6)$$

The amount of horizontal edge after vertical projection is  $h(v)$ . The row index is denoted as  $v$ ,  $\lambda_H$  is the height of the detected vehicle,  $\lambda_s$  is the width of the detected vehicle, and the row of the maximum amount of horizontal edge is  $v^*$ . Vehicle identification is denoted as  $\alpha$ . If the condition is met in (6), this candidate would be considered as a real car and would be tracked in the next frame. The verification of motorcycle is the same as the vehicle verification. The criterion in (6) is important so that most false alarms could be avoided. However, the tradeoff is that some motorcycles with less strong horizontal edges would be deleted.

**3.6. Candidate Matching.** So far, the real cars have been retained. When the full searching mode is processed, there is no idea about the correlation of the same car in consecutive frames. In order to track the detected car in successive frames, the candidate matching function is designed to conquer this problem. Matching candidate function is always executed not only to generate the new trajectory of the detected vehicle or motorcycle in the beginning but also to match the detected vehicle to the tracked vehicle. The first step in this function is finding the closest vehicle in consecutive frames. If the car position is closer in the consecutive frame, some characteristics are verified in (7), where  $H$  and  $W$  represent the height and the width of the image, respectively.  $\beta_C, \beta_L, \gamma_C, \gamma_L, \rho_C$ , and  $\rho_L$ , are the height, center column location, and center row location of the same vehicle in the current frame and last frame, respectively:

$$\begin{aligned} |\beta_C - \beta_L| &\leq \frac{H}{16}, \\ |\gamma_C - \gamma_L| &\leq \frac{W}{8}, \\ |\rho_C - \rho_L| &\leq \frac{3 * W}{32}. \end{aligned} \quad (7)$$

In the end, the trajectory is produced, and the information of the cars has been inherited, updated, and stored. The behaviors of the cars or motorcycles could be determined using the information of the trajectory in tracking mode.

**3.7. Behavior Judgment.** There are three definitions for the behaviors of cars in this system: relative approaching, relative backing, and relative static, respectively. These definitions are

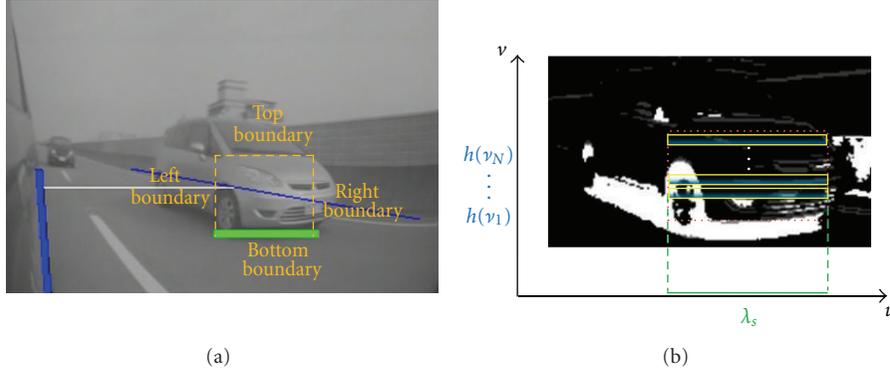


FIGURE 11: (a) Boundary setting for vertical projection. (b) The vertical projection of the horizontal edge.

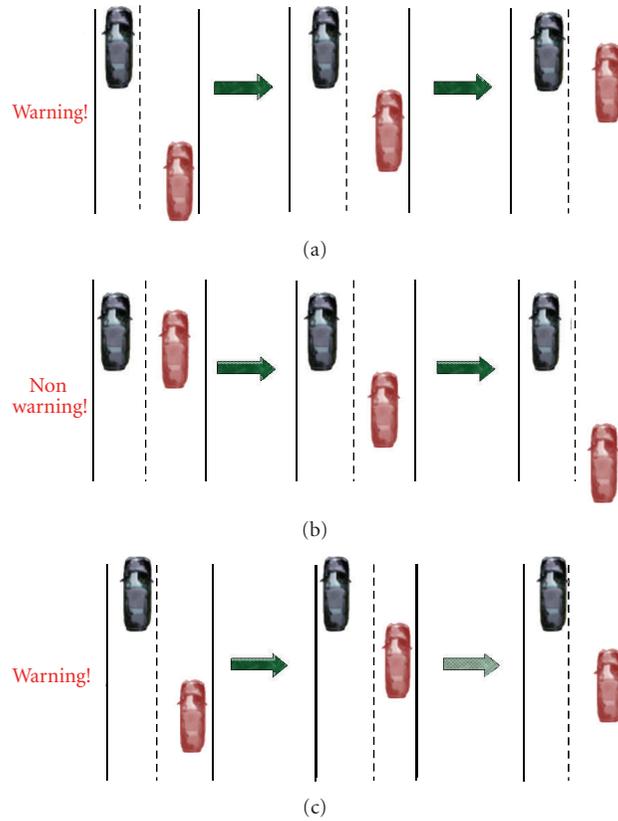


FIGURE 12: (a) Relative approaching. (b) Relative backing. (c) Relative static.

shown in Figure 12. The black car is the host car, and the red car is the tracked car in the blind spot area:

$$\delta_1(t) = v(t) - v(t - 1), \quad (8)$$

$$A(t) = \begin{cases} 1, & \text{if } \delta_1(t) > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

$$RA = \begin{cases} 1, & \sum_{i=t-M-1}^{i=t} A(i) > 0.6 * M, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

$$\delta_2 = |v(t) - v(t - 1)|, \quad (11)$$

$$S(t) = \begin{cases} 1, & \text{if } \delta_2(t) < 3, \\ 0, & \text{otherwise,} \end{cases} \quad (12)$$

$$RS = \begin{cases} 1, & \sum_{i=t-M-1}^{i=t} S(i) > 0.6 * M, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

The relative approaching is computed and judged through (8), (9), and (10), where  $v$  and  $t$  represent the

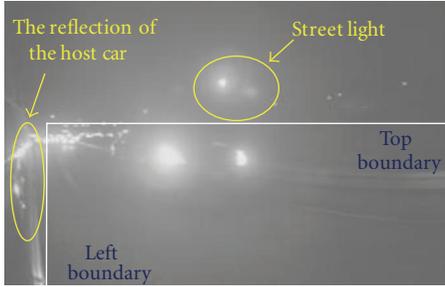


FIGURE 13: The boundary definition of ROI in the nighttime.

bottom position of the tracked vehicle and time index, respectively.  $M$  which is the monitor window is set to 9 here. The relative static is judged by (11), (12), and (13). RA and RS symbolize the relative approach and relative static, respectively. If judgments of both conditions fail, the situation is considered as relative backing. If relative approaching vehicles or relative static vehicles exist, the system should send a warning signal to the driver, otherwise the relative backing vehicles do not have to be warned. Relative approaching and relative backing mean that the speed in the host car is faster or slower than the speed of the tracked vehicle, respectively. Relative static represents that the speed of the host vehicle is similar to that of the tracked vehicle. Because relative static means that there is a tracked vehicle close to the host car, the warning also should be triggered to prevent the collision. Buzzers or LEDs are used for warning signal, which depends on the demands of users.

#### 4. Vehicle Detection Algorithm in the Nighttime

The main feature used in the nighttime is the lamps of vehicles, because most drivers would turn on the lamps in the evening or at night. Therefore, the lamps become the significant feature to identify the vehicle. Before searching the lamps, the ROI for the nighttime has to be determined at first.

**4.1. ROI Definition for the Nighttime Detection.** Unlike the shadow feature under the vehicle, the lamps are always equipped higher than the air dam. Consequently, the lamps would not be within the day time ROI. Therefore, the ROI for the daytime is not suitable for the nighttime. Figure 13 illustrates the ROI definition for the nighttime vehicle detection. The reflection of the host car and the street lamps could be filtered out.

**4.2. Lamp Extraction.** As described above, the lamps within the ROI are the targets to be searched and detected. At first, the system should find out where the lamps appear in ROI. It is supposed that the bright objects might be the lamps in ROI, so extracting the bright objects in ROI is the first step in image preprocessing for the nighttime. Calculating the threshold in (14) to extract this feature, we get statistic of the gray level value of pixels in ROI and establish a histogram

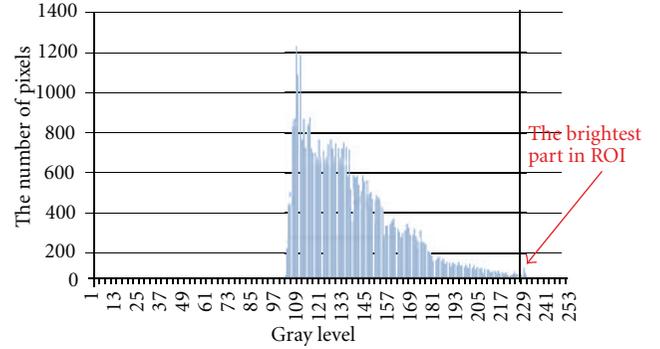


FIGURE 14: The gray level histogram in ROI in the nighttime.

for this frame, as shown in Figure 14. The 1% brightest part in ROI is considered as bright objects:

$$g^* = \arg \left( \sum_{g=0}^{255} h(g) > \delta N \right). \quad (14)$$

The amount at gray level  $g$  is denoted as  $h(g)$ , and the calculated threshold in this frame is  $g^*$ .  $N$  is the number of pixels in the nighttime ROI, and  $\delta$  is 0.99 here. The sizes of bright objects might be seriously affected if the dynamic threshold changed too fast. In order to avoid the dynamic threshold changing severely, the past thresholds would be referenced in this frame, as in (15). The calculated threshold in this frame and in the last frame are  $g^*(t)$  and  $g^*(t-1)$ , respectively. The used threshold in the current frame is  $g(t)$ , where  $t$  is time index:

$$g(t) = \frac{7}{8}g^*(t) + \frac{1}{8}g^*(t-1). \quad (15)$$

The bright pixels would be extracted if their gray level is higher than the threshold. Then, to obtain the clean feature plane, the erosion is performed to remove the noise, and the result is displayed in Figure 15. After that, the connected component labeling is performed to save the rectangle information of the bright objects.

**4.3. Lamp Verification.** Although the bright objects mentioned above are extracted, some false bright objects caused from the reflection of the pavement and the island would be included. Therefore, we have to filter out the wrong lamp candidates in this flow. First, the intensity variance and size judgments of the bright objects would be performed. The intensity mean and variance are calculated in (16) and (17), where  $I_i$  is the gray level value of the  $i$ th pixel of this bright object,  $M$ ,  $\bar{I}$ , and  $S$  are the total number of pixels, intensity mean, and variance of this bright object. After that, the width and height of the bright object should meet the requirements

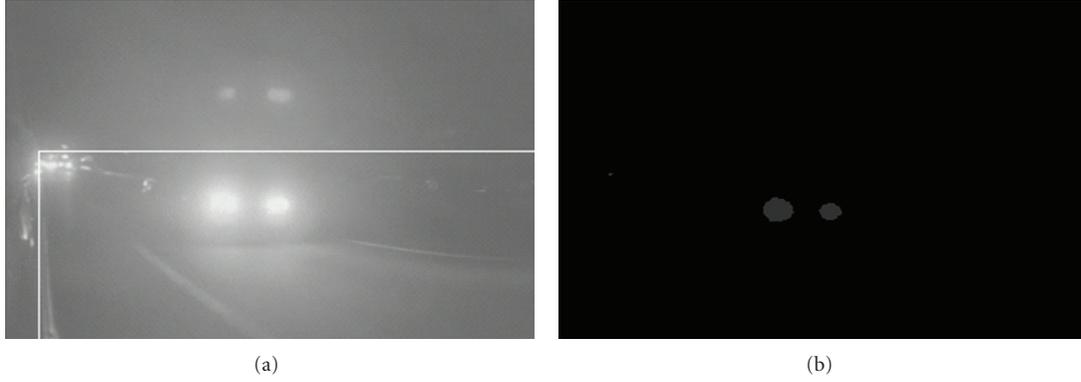


FIGURE 15: (a) The source image in the nighttime. (b) The extracted bright object plane of the left image.

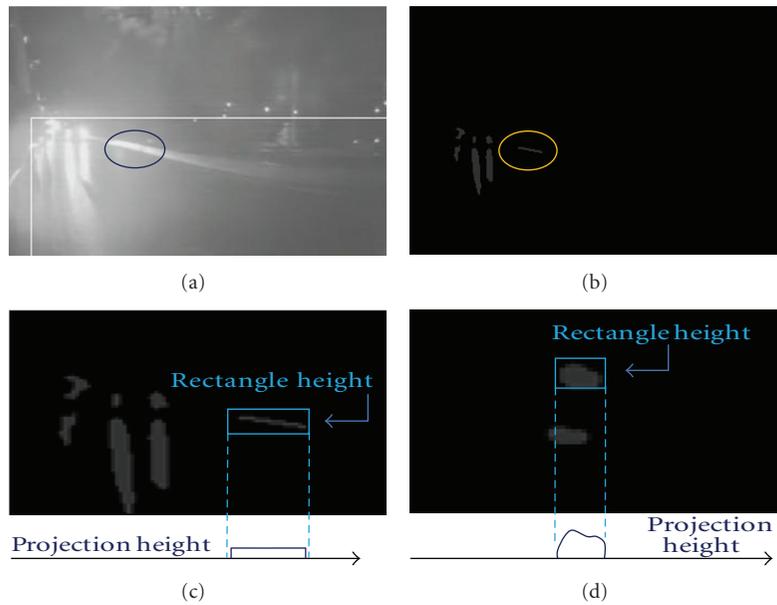


FIGURE 16: (a) The source image with the reflection of the safety island at night. (b) The extracted bright object plane. (c) Vertical projection of the bright object on the safety island. (d) Vertical projection of the bright object of the lamp.

in (18) and (19).  $H_k$ ,  $W_k$ ,  $H$ , and  $W$  are the height and width of the  $k$ th bright object in the image, respectively,

$$\bar{I} = \frac{1}{M} \sum_{i=1}^M I_i, \quad (16)$$

$$S = \left( \frac{1}{M} \sum_{i=1}^M I_i^2 \right) - \bar{I}^2, \quad (17)$$

$$\frac{H}{80} < H_k < \frac{H}{4}, \quad (18)$$

$$\frac{W}{60} < W_k < \frac{W}{4}. \quad (19)$$

The reflection of bright objects on the pavement caused by rainy day could be filtered out directly through the width, height, and area judgments. However, when driving through the urban ways, the bright object caused by safety island

appears frequently in the image of the left side of the camera, as shown in Figure 16. Because the rectangle size of safety island bright object is similar to the lamp object, the vertical projection of the bright object is operated to filter out the reflection bright object. The heights of the safety island after vertical projection would be lower than the heights of the general lamps after projection, as shown in Figure 16. So, if the projection height is lower than the half of the rectangle height, this object must be considered as a reflection bright object on the safety island and it would be filtered by this method. Otherwise, this bright object would be retained and considered as a lamp object.

**4.4. Filter the Lamps in the Second Next Lane.** Since we only focus on the vehicle or motorcycle in the next lane, this subsection is used to judge and filter the lamps appearing in the second next lane. Because of the image captured by the camera lacks the depth information in 2D plane, the



FIGURE 17: (a) The lamps belong to the vehicle in the second lane which are not the targets for this system. (b) The lamp belongs to the locomotive in the next lane.

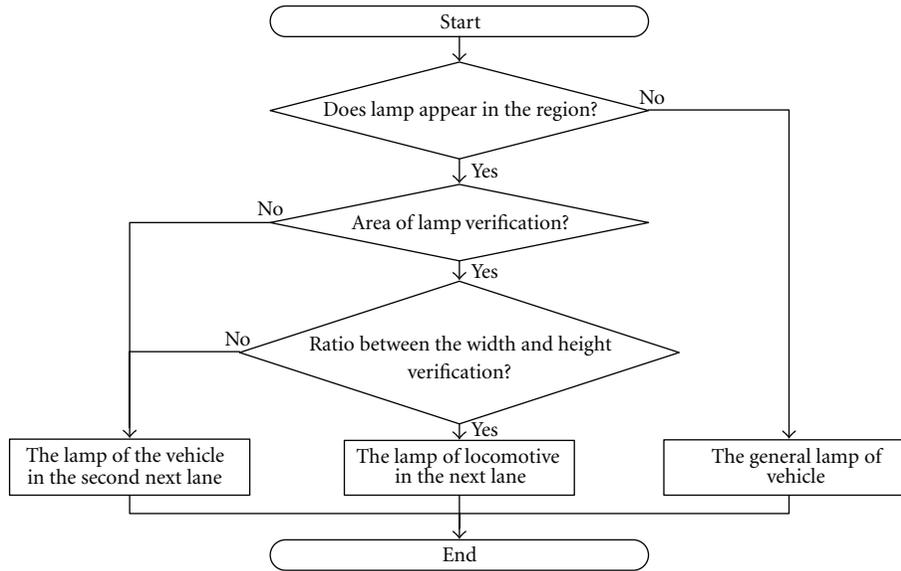


FIGURE 18: The flow of advanced lamp verification.

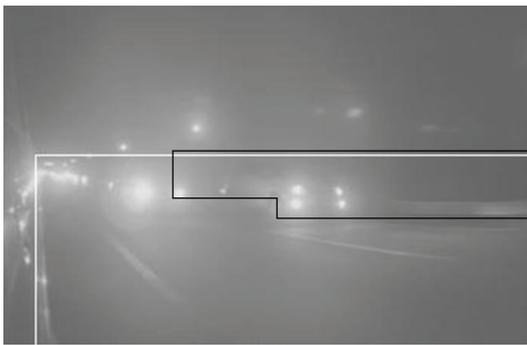


FIGURE 19: The definition of the region which might be in the second next lane.

positions of lamps from the vehicle in next two lanes and from the locomotive in next lane would appear in the same region, as shown in Figure 17. If the lamp belongs to the locomotive in the next lane, it needs to be kept tracking. However, if the lamp belongs to the vehicle in the second next lane, it is not the target to be tracked in this system.

Figure 18 illustrates the flow of lamps advanced conditions judgment, and the definition of the region of the second next lane is shown in Figure 19. If a bright object appears in this region, it might be the lamp of the vehicle in the second next lane, and the area and ratio verification would be processed in (20).  $\Phi_A$ ,  $\Phi_W$ , and  $\Phi_H$  are the thresholds of area, width, and height of the lamp rectangle, respectively. 150, 9, and 7 are chosen here in this system.  $\zeta$  which is the threshold of the ratio between the width and height is 0.25:

$$\begin{aligned}
 A(k) &> \Phi_A, \\
 W(k) &> \Phi_W, \\
 H(k) &> \Phi_H, \\
 \frac{W(k)}{H(k)} &> \zeta.
 \end{aligned} \tag{20}$$

**4.5. Lamps Tracking.** This part would only be processed in the tracking mode. It is used to build the relations between lamps in consecutive frames. The concept in this part is identical to match candidates. The first step is checking conditions between lamps, as in (21), (22), and (23).  $H_C$ ,  $W_C$ ,  $H_L$ , and  $W_L$  are the height and width of the same lamp in

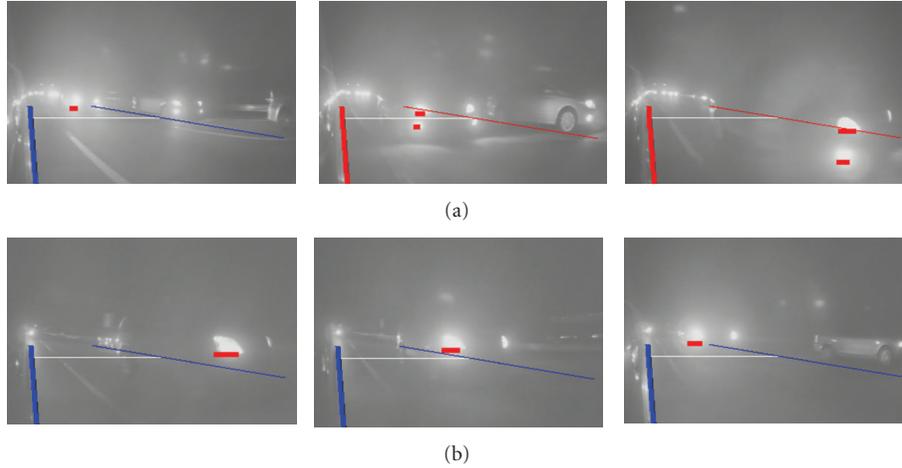


FIGURE 20: (a) Relative approaching. (b) Relative backing.

the current frame and in the last frame, respectively.  $T_H$ ,  $T_W$ ,  $U_R$ , and  $L_R$  are the thresholds for these conditions, and 10, 10, 1.35, and 0.7 are chosen for them here, respectively. If the conditions mentioned above are matched and the computed distance in (24) is minimal, the trajectory of this lamp is generated. The distance between the  $i$ th lamp in the current frame and the  $j$ th lamp in the last frame is  $d_{ij}$ , and the column and row of center coordinates of the lamp rectangle are denoted as  $x$  and  $y$ :

$$|H_L - H_C| < T_H, \quad (21)$$

$$|W_L - W_C| < T_W, \quad (22)$$

$$L_R \leq \frac{H_L/W_L}{H_C/W_C} \leq U_R, \quad (23)$$

$$d_{ij} = |x_i - x_j| + |y_i - y_j|. \quad (24)$$

**4.6. Behavior Judgment for the Nighttime.** The lamps usually move from left to right or from right to left in images in the nighttime. The vertical movements of lamps are not obvious, so column information is more important for behavior judgment here. The judgment is same to the judgment at daytime in Section 3, but the input parameter changes from the row index to the column index of the trajectory. Figure 20 depicts the output image sequence from left to right.

## 5. Experimental Results

This system has been on-road tested through the long verification. Several challenge video sequences are tested and the results are illustrated in Figure 21. If ROI is drawn in red color, the status represents that the system detected and tracked a vehicle which is relative approaching or relative static in ROI. Otherwise, ROI would be drawn in blue color. The buzzers or LEDs could be the warning signal to the

drivers through the GPIO on DSP, instead of red lines. In general, the proposed system could detect and track the vehicle, bus or motorcycle exactly in the daytime and in the nighttime. Moreover, the system can detect vehicles, even if the reflection is serious at rainy night. For the quantitative evaluation of the vehicle and the motorcycle detection performance, detection ratio, and false alarm ratio commonly used for evaluating performance in information retrieval [19] are adopted in this study. The measures are defined as:

$$\begin{aligned} DR &= \frac{T_p}{T_p + F_n}, \\ FAR &= \frac{F_p}{T_p + F_p}, \end{aligned} \quad (25)$$

where  $T_p$  (true positives) represents the number of correctly detected vehicles,  $F_p$  (false positives) represents the number of falsely detected vehicles, and  $F_n$  (false negatives) is the number of the missing vehicles.

Table 1 exhibits the quantitative results of the proposed approach for the vehicle and motorcycle detection and tracking, including the testing time, the detection ratio, and the false alarm ratio on high-speed road (H) and urban road (U) in sunny days and rainy days. The detection ratio achieves to 94%. The false alarm ratio in rainy days are much higher than others because that the camera was set up out of the car. When driving in rainy day, the lens on the camera is easily wet by rainwater. Therefore, this problem would lead to a lot of false alarms. The false alarm ratio is lower than 9.41% except rainy days. The following part evaluates the performance of the proposed system and compares it to the region-based method [17] and the edge-based method [18]. Both of these methods are implemented on DSP platform, so they could be compared here on TI DM6437. Table 2 shows the experiment data of the both methods and our systems. Sobel edge is the only feature used in [18], so that there are no results for the nighttime. Some representative comparative results of vehicle detection for the challenge sequences by

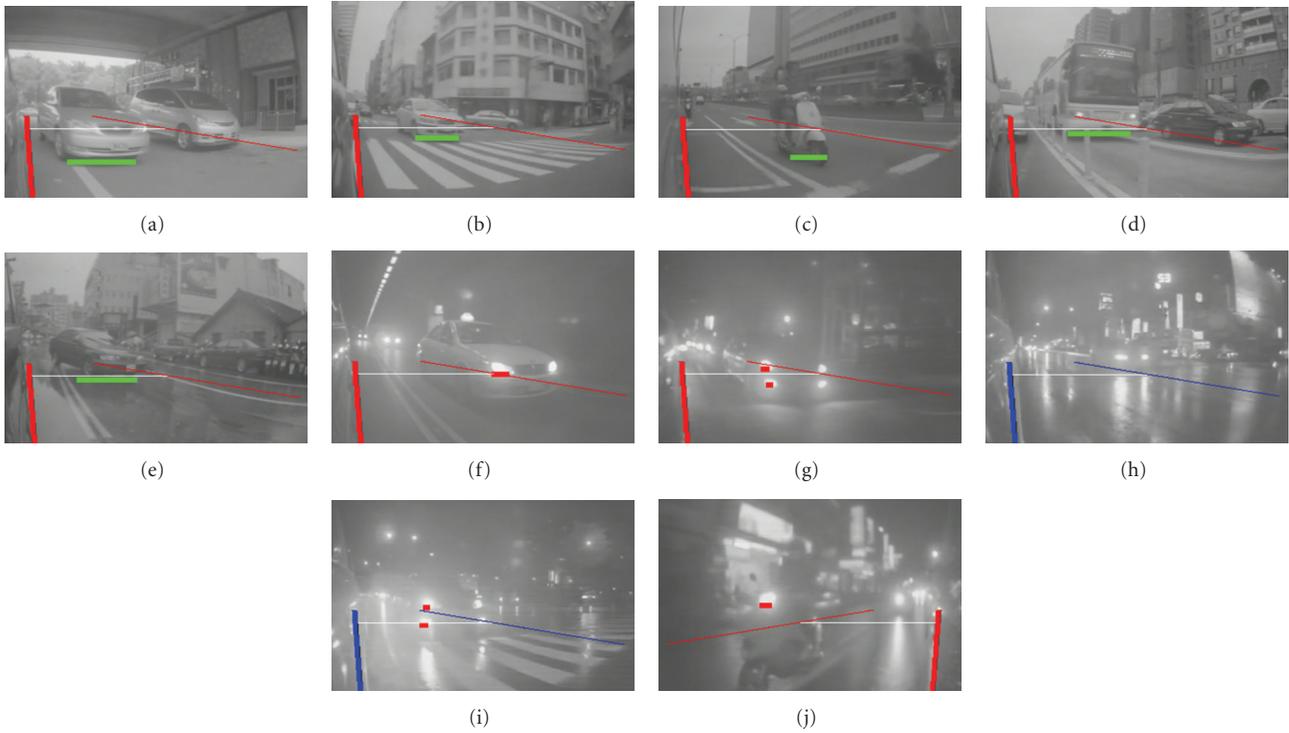


FIGURE 21: Parts of output images to show the experiment results in complex environment. (a) The general cars on complex urban road in the daytime. (b) The tracked vehicle across the zebra crossing. (c) The locomotive on urban road in the daytime. (d) The bus on urban road in the daytime. (e) The car on urban road in rainy day. (f) The car in the tunnel. (g) The general cars in the nighttime. (h) The reflection bright objects on the pavement caused by rainy night. (i) The general car on urban road at rainy night. (j) The locomotive on urban road at rainy night.

TABLE 1: The experimental data of the proposed approach.

Time (minutes)	Daytime				Nighttime			
	Sunny		Rainy		Normal		Rainy	
	H	U	H	U	H	U	H	U
	128	238	105	65	200	145	90	90
$T_p$	100	548	72	77	95	372	43	131
$F_n$	0	36	0	4	4	9	0	8
$F_p$	2	53	1	8	7	36	25	31
DR (%)	100	93.84	100	95.06	95.96	97.64	100	94.24
FAR (%)	1.96	8.82	1.37	9.41	6.86	8.82	36.76	19.14

TABLE 2: The experiment data for comparison.

Methods	C. T. Chen and Y. S. Chen [17]				Çayir and Acarman [18]		Proposed			
	10 (ms)				16 (ms)		17 (ms)			
	Daytime		Nighttime		Daytime		Daytime		Nighttime	
Time	H	U	H	U	H	U	H	U	H	U
Road										
Test minutes	30	60	30	28	30	60	30	60	30	28
DR (%)	100	100	100	100	82.35	77.27	100	95.45	100	96.15
FAR (%)	20.93	56	41.54	54.73	22.22	22.73	0	0	0	7.41

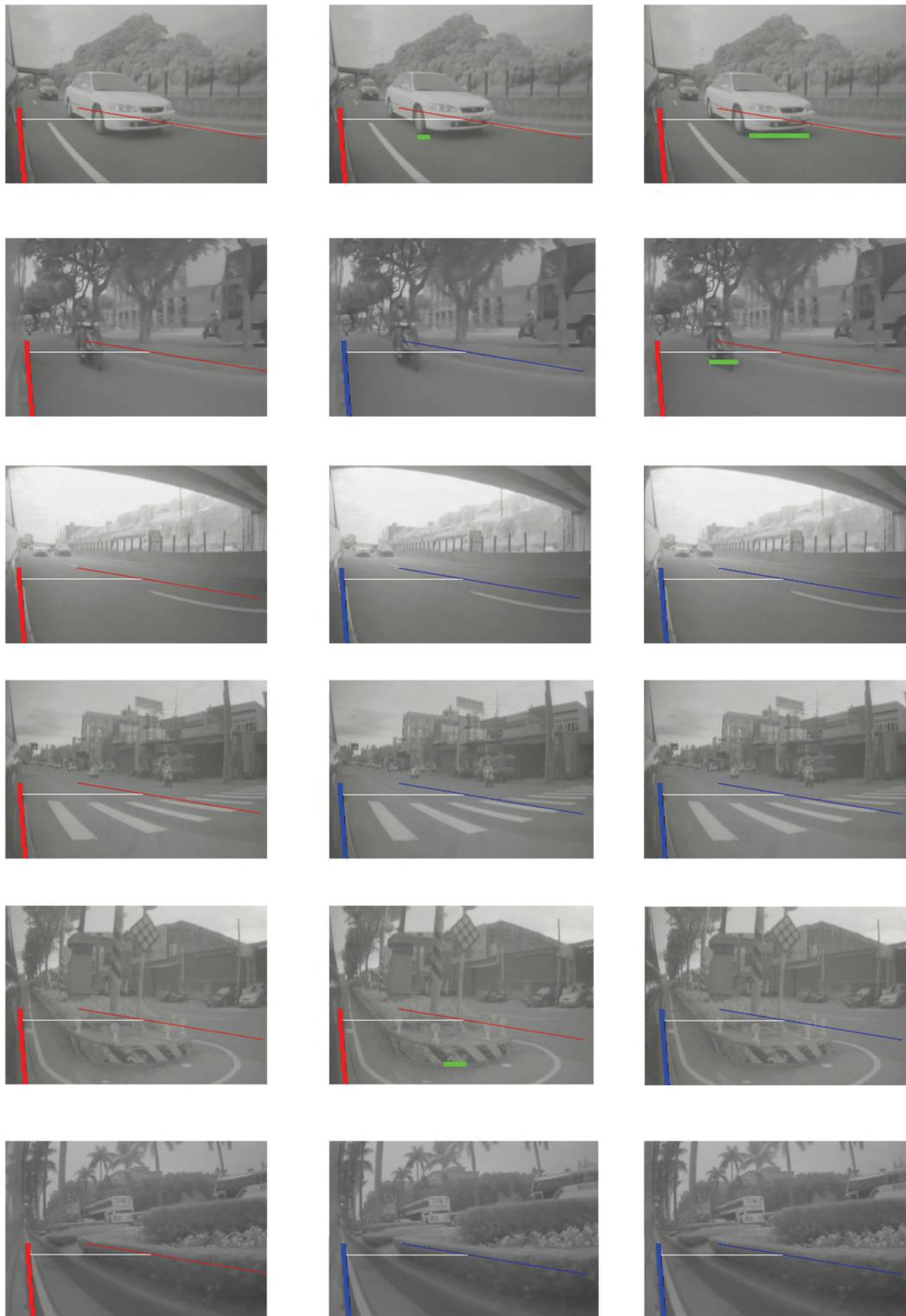


FIGURE 22: Comparative results. First column is the region-based by C. T. Chen and Y. S. Chen [17]. Second column is the edge-based by Çayır and Acarman [18]. Third column is the proposed method. First row: vehicle in general. Second row: motorcycle in general. Third row: shadow under the bridge. Fourth row: zebra crossing marking. Fifth row: safety island. Sixth row: haystack.

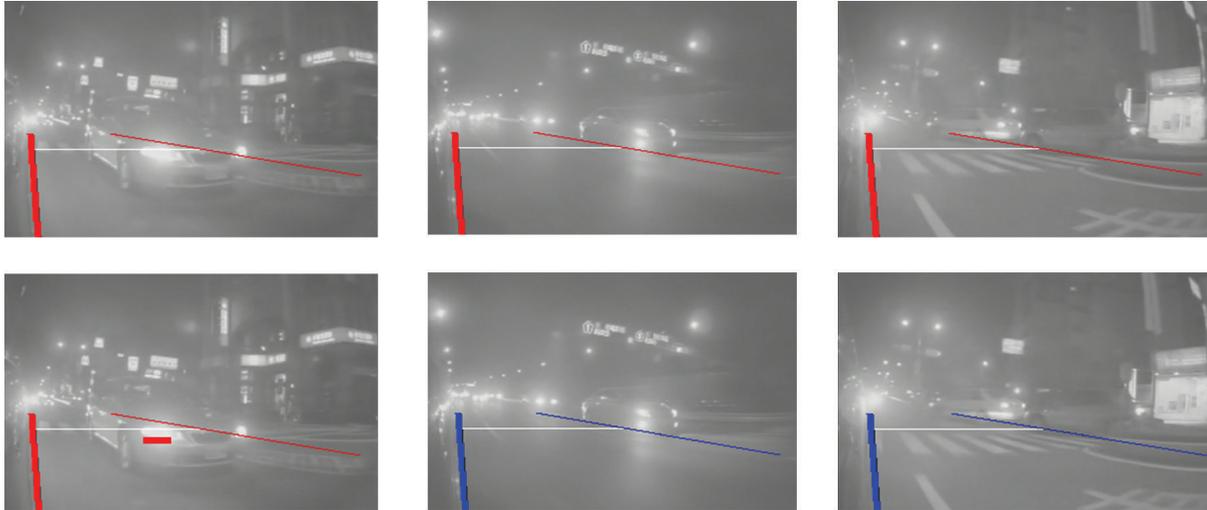


FIGURE 23: Comparative results in the nighttime. First row is the region-based by C. T. Chen and Y. S. Chen [17]. Second row is the proposed method. First column: vehicle in general. Second column: vehicle in the second next lane. Third column: lights of the store.

the proposed approach, the region-based method, and edge-based method are illustrated in Figures 22 and 23 and Table 2. The detection ratio is always 100% in [17], but false alarm ratio is also very high, at least 20.93%. The detection ratio of [18] is not better than others, and the false alarm ratio is not lower than the method of the proposed method, too. The processing time of the proposed system in each frame with the optimized code is about 17 (ms).

## 6. Conclusions

The proposed approach presents a real-time embedded blind spot safety assistance system to detect the vehicle or motorcycle appearing in the blind spot area. First, algorithms for daytime and nighttime are both implemented in our system. Second, the automatic threshold approach for the shadow and the bright object segmentation is used to overcome the problem of varying light conditions in outdoor scenes. Next, edge feature method is utilized to remove the noises in the complex environments. Then, lamp verification is used to distinguish between the headlight and the reflection, and the advance lamp verification is used to filter the vehicle in next two lanes. Tracking procedure is applied to analyze the spatial and temporal information of the vehicle, and the behavior judgment is judged by the vehicle trajectory. This algorithm solves most of problems caused in various kinds of environments, especially in the urban. It also overcomes the weather conditions and keeps the high performance no matter what time it is. The proposed algorithms were also implemented on a TI DM6437 and tested with the real highway and the urban road in the daytime and nighttime. From the experimental results, it is obvious that the proposed approach not only works well in highway but also has good performance in the urban. With comparing with other solutions, the experimental results also show that the proposed approach has better performance both in detections and false alarms.

## Acknowledgment

This research was supported by National Science Council, under Grant NSC 100-2221-E-009-041.

## References

- [1] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, 2006.
- [2] W. C. Chang and C. W. Cho, "Online boosting for vehicle detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 3, pp. 892–902, 2010.
- [3] R. O'Malley, E. Jones, and M. Glavin, "Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, Article ID 5446402, pp. 453–462, 2010.
- [4] X. Liu and K. Fujimura, "Pedestrian detection using stereo night vision," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1657–1665, 2004.
- [5] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert, "Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner," *Journal of Autonomous Robots*, vol. 19, no. 2, pp. 117–140, 2005.
- [6] C. Y. Wong and U. Qidwai, "Intelligent surround sensing using fuzzy inference system," in *Proceedings of the 4th IEEE Conference on Sensors 2005*, pp. 1034–1037, November 2005.
- [7] O. Achler and M. M. Trivedi, "Vehicle wheel detector using 2D filter banks," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 25–30, June 2004.
- [8] M. Ruder, W. Enkelmann, and R. Garnitz, "Highway lane change assistant," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, vol. 1, pp. 240–244, 2002.
- [9] J. Díaz, E. Ros, S. Mota, G. Botella, A. Cañas, and S. Sabatini, "Optical flow for cars overtaking monitor: the rear mirror blind spot problem," in *Proceedings of the 10th International Conference on Vision in Vehicles*, pp. 1–8, Granada, Spain, 2003.

- [10] P. H. Batavia, D. A. Pomerleau, and C. E. Thorpe, "Overtaking vehicle detection using implicit optical flow," in *Proceedings of the International IEEE Conference on Intelligent Transportation Systems (ITSC '97)*, pp. 729–734, November 1997.
- [11] B. E. Stuckman, G. R. Zimmerman, and C. D. Perttunen, "A solid state infrared device for detecting the presence of car in a driver's blind spot," in *Proceedings of the 32nd Midwest Symposium on Circuits and Systems*, pp. 1185–1188, August 1989.
- [12] M. Krips, J. Velten, A. Kummert, and A. Teuner, "AdTM tracking for blind spot collision avoidance," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 544–548, June 2004.
- [13] T. Yoshioka, H. Nakaue, and H. Uemura, "Development of detection algorithm for vehicles using multi-line CCD sensor," in *Proceedings of the International Conference on Image Processing (ICIP '99)*, pp. 21–24, October 1999.
- [14] A. Techmer, "Real-time motion analysis for monitoring the rear and lateral road," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 704–709, June 2004.
- [15] K. Furukawa, R. Okada, Y. Taniguchi, and K. Onoguchi, "Onboard surveillance system for automobiles using image processing LSI," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 555–559, June 2004.
- [16] S. Jeong, S. W. Ban, and M. Lee, "Autonomous detector using saliency map model and modified mean-shift tracking for a blind spot monitor in a car," in *Proceedings of the 7th International Conference on Machine Learning and Applications (ICMLA '08)*, pp. 253–258, December 2008.
- [17] C. T. Chen and Y. S. Chen, "Real-time approaching vehicle detection in blind-spot area," in *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (ITSC '09)*, pp. 1–6, October 2009.
- [18] B. Çayır and T. Acarman, "Low cost driver monitoring and warning system development," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 94–98, June 2009.
- [19] I. Cohen and G. Medioni, "Detecting and tracking moving objects for video surveillance," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 319–325, June 1999.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

