Expert Systems with Applications 37 (2010) 113–126

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Applying fuzzy method to vision-based lane detection and departure warning system

Jyun-Guo Wang [a], Cheng-Jian Lin [b,*], Shyi-Ming Chen [a]

[a] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung County 411, Taiwan, ROC

ARTICLE INFO

Keywords:
Self-clustering algorithm
Fuzzy C-mean
Lane detection
Lane departure warning system

ABSTRACT

As the high growth of population of vehicles, the traffic accidents are becoming more and more serious in recent years. Most occurrences of the car accidents results from the distraction, inattention and driving fatigue of the driver. Hence, in order to avoid the driver being in danger as much as possible. In the lane detection, in order to enhance lane boundary information and to suitable for various light conditions all day, we combine the self-clustering algorithm (SCA), fuzzy C-mean and fuzzy rules to process the spatial information and Canny algorithms to get good edge detection. In the lane departure warning, the system uses instantaneous information from the lane detection to calculate angle relations of the boundaries. The system sends a suitable warning signal to drivers, according to degree different of the departure. These experiments have been successfully evaluated on the PC platform of 3.2-GHz CPU and the average frame rate is up to 14 fps.

1. Introduction

Intelligent Transportation System (ITS) is a system that applies advanced technologies such as electronics, communication, information, image processing, and various sensors to catch real-time information that helps not only to improve transportation safety and mobility, enhance productivity, but also reduce transportation impacts on the environment. According to the website of the US Department of Transportation's (USDOT) ITS program, the applications of ITS can generally be divided into two parts, intelligent infrastructure systems and intelligent vehicle systems. Systems such as arterial management, freeway management, traveler information, information management, emergency management, electronic payment, etc. are intelligent infrastructure applications. Other systems, such as collision avoidance systems, and driver assistance systems, are intelligent vehicles applications of. Similar to the USDOT ITS program, other countries such as Japan and Taiwan also includes the Advanced Vehicle Control and Safety System (AVCSS) in their ITS field. Among these ITS applications, we can easily conclude that car safety is one of the most important issues in many countries.

Several researchers around the world have been developing vision-based ASV systems for lane detection, lane following and lane departure warning; however, most of them present limitations in situations involving shadows, varying illumination conditions, bad conditions of road paintings and other image artifacts.

Chen, Jochem, and Pomerleau (1995) developed a roadway departure warning system by setting a downward-looking video camera to monitor the vehicle's lateral displacements. LeBlanc et al. (1996) proposed a road-departure prevention system that predicts the vehicle's path and compares such path with the sensed road geometry to estimate the time to lane-crossing (TLC); however, the vision-based sensor requires good lighting and pavement conditions to detect lane boundaries. Kwon et al. (1999) developed a vision-based lane departure warning system that considers two warning criteria: the lateral offsets and the time to lane-crossing (TLC). With some considered states of giving alarm or giving low level warning, they divided the warning level into three levels: give an alarm state, give a low level warning state, and give no alarm state. Risack, Mohler, and Enkelmann (2000) proposed a lane keeping assistance system, which warns the driver on unintended lane departures. In fact, they used an existing video-based lane detection algorithm and compared different methods to detect lane departure, using several assumptions on driver behavior in certain situations to distinguish between intended and unintended lane departures. Lane departures are successfully detected by their technique, but they also needed roads in good conditions and good lighting conditions. Lee (2002) proposed a lane departure detection system that estimates lane orientation through an edge distribution function (EDF), and identifies changes in the traveling direction of a vehicle. However, the EDF may fail in curved roads with dashed lane markings. A modification of this technique (Lee, Kee,

[*] Corresponding author.
E-mail address: cjlin@ncut.edu.tw (C.-J. Lin).

0957-4174/$ - see front matter Crown Copyright © 2009 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.05.026

& Yi, 2003) includes a boundary pixel extractor to improve its robustness. However, curved lanes may still cause problems, because a linear model (computed using the Hough transform) is used for fitting lane boundaries. Hsu, Cheng, Tsuei, and Huang (2002) developed a LDWS that considers the angle of lane boundaries and adaptively measured both of the angles and their differences to see the vehicle's motion. The processing speed was 10 frames/s. Apostoloff and Zelinsky (2003) proposed a lane tracking system based on particle filtering and multiple cues. This method does not track explicitly the lanes, but it computes parameters such as lateral offset and yaw of the vehicle with respect to the center of the road. Although the method appears to be robust under a variety of conditions (shadows, different lighting conditions, etc.), it cannot be used to estimate curvature or detect if the vehicle is approaching a curved part of the road. Hsu (2003) applied radial basis probability networks (RBPN) as the pattern recognition mechanism that measures and records vehicle's lateral displacement and its rate of change, and compare the trajectory with the training patterns to find one classification that fits most and see if the vehicle is about to lane departure. Three warning levels: warning, caution, and safety were considered. McCall and Trivedi (2004) proposed a method for lane detection using steerage filters. Such filters perform well in picking out both circular reflector road markings as well as painted line road markings. Filter results are then processed to eliminate outliers based on the expected road geometry and used to update a road and vehicular model along with data taken internally from the vehicle. Such a technique is robust with respect to lighting changes and shadows, but has shortcomings for relatively curved roads (because this method relies basically on a linear model). Jung and Kelber (2004) developed a lane departure warning system based on a linear-parabolic lane model. They divided the road ahead the vehicle into two filed: near and far. A linear function is used to fit the near vision field, and a quadratic function is used for the far field. The warning algorithm of Jung and Kelber was comparison of the orientation of both left and right lane boundaries. In order to reduce the influence of noise, the orientations were averaged in five consecutive frames in this study (15 frames/s). Jung and Kelber (2005) also provided a LDWS using lateral offset (LO) with uncalibrated camera that takes into account the lateral offset and its rate of change were considered.

On the basis of several researchers worldwide have been developing vision-based ASV systems. Almost of methods proposed before are only suitable for particular weather situation. Consequently, they do not adapt to real environment. Therefore, in order to improve the situation describe above. We proposed our methods to solve all kinds of weather and we aim at developing a Drive Assistance System (DAS) that can help the driver to be aware of dangerous situations which may have been ignored by the driver due to inattention or fatigue. As stated earlier, in this research, we focus to solve two problems- lane detection and roadway departure – under all kinds of weather.

The organization is listed as follows. In Section 2, the camera configuration and system framework is first introduced. The lane boundary detection procedure is developed in Section 3, and the lane departure warning is presented in Section 4. In Section 5, the experiments results will be shown. The conclusions are shown in the last Section.

2. Camera configuration and system framework

In this Section, the fundamentals of this research are discussed. A vision-based driver assistance system usually needs calibration before operating. We want to build an easy-to-install driver assistance system, so that it can be used without knowing the details of the configuration. If the system was commercialized, customers would be able to use it directly without reading manual.

In the literature, it is quite popular to use vanishing point detection for camera calibration. Camera calibration is meant to determine the parameters of transformation between the image coordinate and the vehicle coordinate. There are many approaches to the task of camera calibration. The most typical one, such as DLT method (Abdel-Aziz & Karara, 1971), is to model the transformation as parametric linear transformation, and then solve the parameters based on a set of known points in the world coordinate system and their corresponding points in the image coordinate system. The main shortcoming of this approach is inconvenient to physically measure a larger quantity of points in the world coordinate system. Instead of taking data of numerous points in world coordinate system, some approaches (Echingo, 1990; Wang & Tsai, 1991) use calibration targets that has some known geometric properties. In this case, the main drawback is however that calibration target must be precisely installed. Another related research (Bas & Crisman, 1997) is to measure the height and the tilt of the

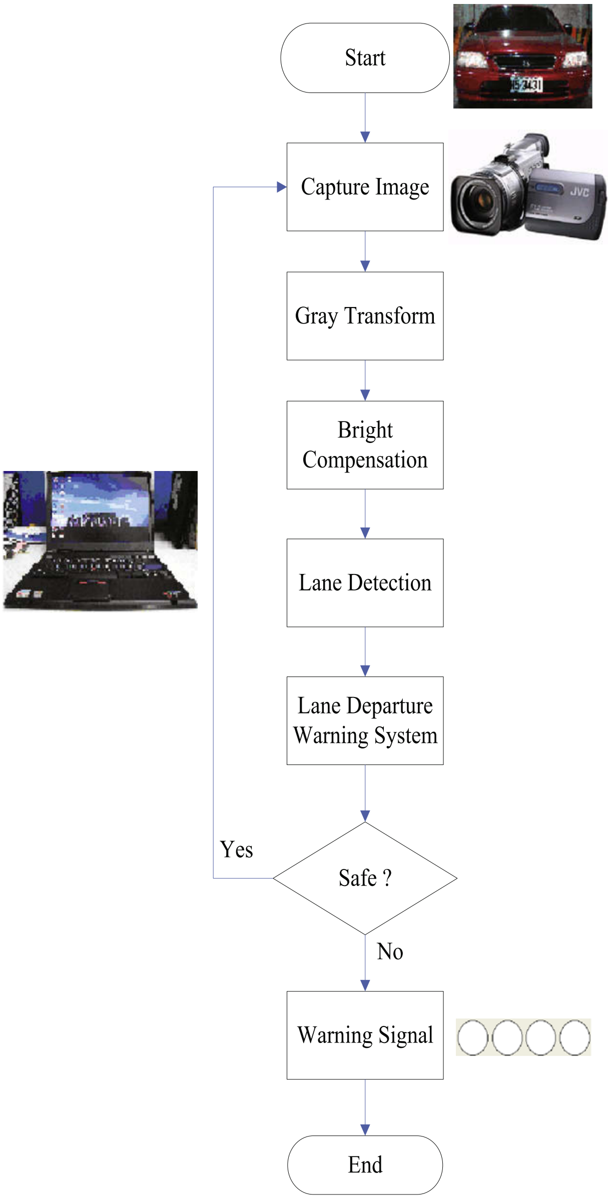


Fig. 1. The flow chart of system operation.

camera, which plus the information of the given vanishing point allows one to compute the focal length and the pan of the camera.

According to the above research, the camera calibration facilitates the transformation of the image coordinates and the world coordinates. Through the calibration, we can acquire the lateral and longitudinal information in the real world coordinate system. In addition, this work applies the method developed by Thomas (2000), to modify the calibration experiment results with mathematical formulas. Based on the modified information, this research further applies the α–β filter and the simple moving average method to dynamically predict the relative distance and velocity, both laterally and longitudinally. The objectives to camera calibration are to transform the image system captured from the CCD camera to the real world system and to build the transformation mechanism between the image and world coordinates. First, we have an experimental vehicle equipped with a JVC GR-DV4000U CCD camera mounted on the front windshield inside the experimental vehicle that captures front view images. The size of the captured image is 320 × 240 pixels. Second, an industrial personal computer placed in the trunk of the vehicle, with an image catching program developed for this study.

To achieve these two objectives, a single camera is installed on a real vehicle to grab the scenes from a real traffic environment for lane detection and roadway departure application under all kinds of weather. Fig. 1 shows the proposed system operation conceptual flowchart. In the locating phase, it consists of two procedures; lane detection and roadway departure estimates. Finally, the detected lane boundaries and roadway departure estimated will be used to detect the potential dangerous situations via hazard identification procedure. If the driver is in the dangerous situations, an automatic voice system or warning signal system will warn the driver via a loud speaker or warning signal color degree, according to various departure situations.

3. Lane detection

At the beginning of this architecture, the RGB coordinate will be transformed into the YCbCr one so that the illumination component will be totally retained because we only require monochromatic information of each frame to process the data. Then, the automatic brightness compensation about deal with of the image content will be described in Section 3.1. The preprocessing step for edge detection will be presented in Section 3.2. Next to the processing step, we used fan scanning detection to get border lines of possible vehicles, as described in Section 3.3.

3.1. Automatic brightness compensation using fuzzy rules

Before discussing how to search for the lane-marking, the step of color transformation must be executed. In general, most of the algorithms shown in the past theses with respect to lane detection have only considered the grey-level component. This reason is that the contrast between the lane boundary and the normal road plane can be easily seen by normal people as usual even if the colors of lanes are not necessarily the same. As a result, the information of luminance for each frame must be stored in our system using the RGB-to-YIQ transformation. On the other hand, the remaining chrominance components such as I and Q are not taken seriously due to the insensitive perception about human eyes. The formulation of transformation can be described by Eq. (1)

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

Based on previous research, we know that the brightness variations will have an impact on the analysis of lane line. To improve this situation, we first calculate each as the pixel of the accumulated value in each gray level image and calculate of each pixel of the accumulated value by way of intensity profile of roadway part block in the image. Then, we have to calculate the center of the area (COA) in the entire image. The formulation of calculate can be described by Eq. (2), where $i = 1, 2, \ldots, 255$ denotes the gray-level, H_i denotes accumulation of pixel.

$$COA = \frac{\sum_{i=0}^{i=255}(H_i \times i)}{\sum_{i=0}^{i=255} H_i} \tag{2}$$

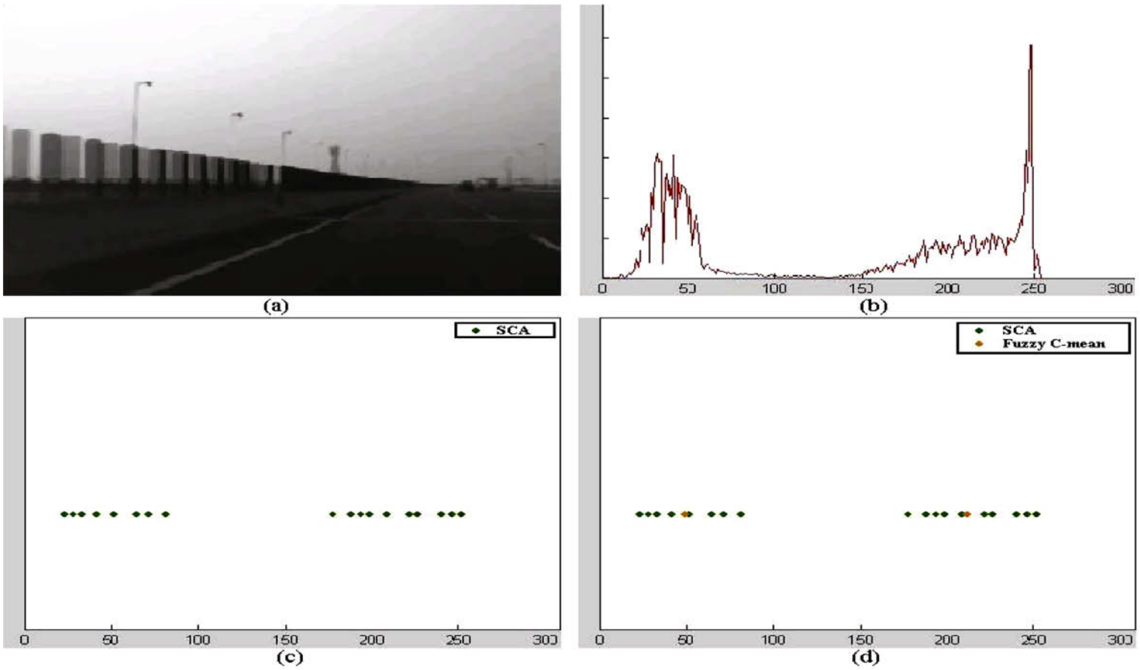


Fig. 2. (a) The gray image; (b) the histogram of gray image; (c) after using the SCA cluster and (d) after using the fuzzy C-mean cluster.

To continue, we calculate two distribution values that separate out the light component and dark component by way of intensity profile of the roadway part block of the image. We use the self-clustering algorithm (SCA) (Lin & Xu, 2006) and fuzzy C-mean (John & Langari, 1999) to analyze the cumulative value of each pixel on the histogram. The transformation in data is illustrated by Fig. 2 and the detailed procedures are as follow:

The proposed SCA is a one-pass algorithm that dynamically estimates the number of clusters in a dataset and finds their means in the input data space. For the above reason, the SCA can cluster the input space quickly. The details of the SCA algorithm are described in the following steps:

- *Step* 0: Create the first cluster C_1 by simply taking the position of the first input data as the first cluster mean *Cci_1* where *i* means the *i*th input variables. Set a dimension distance value to 0 (see Fig. 3a).
- *Step* 1: If all of the training input data have been processed, the algorithm is finished. Otherwise, the current input example, $x_i[k]$, is taken and the distances between this input example and all already created cluster mean *Cci_j* are calculated:

$$D_{i\_j}[k] = \|x_i|[k] - Cc_{i\_j}\|, \tag{3}$$

  where $j = 1, 2, \ldots, R$ denotes the *j*th cluster, $k = 1, 2, 3, \ldots, N$ represents the *k*th input, and $i = 1, 2, \ldots, n$ represents the *i*th dimension.
- *Step* 2: If the distance calculation in Eq. (3) is equal to or less than all of the dimension distances *CDi_j* that represent the *i*th dimension distance in the *j*th cluster (set to 0 initially), then the current input example belongs to a cluster with the minimum distance:

$$D\min_{j}[k] = \min\left(\sum_{i=1}^{n} \|x_i[k] - Cc_{i\_j}\|\right), \tag{4}$$

$$D\min_{i\_j}[k] = \|x_i|[k] - Cc_{i\_j}\|, \tag{5}$$

  where *j* in Eq. (5) represents the *j*th cluster that is computed using Eq. (4). The use of Eqs. (4) and (5) is to find the minimum sum of all the dimension distances in a cluster with the *k*th input data. The constraint is described as follows:

$$D\min_{i\_j}[k] \leqslant CD_{i\_j}, \tag{6}$$

  If no new clusters are created or no existing clusters are updated (the cases of $(x_1[4], x_2[4])$ and $(x_1[6], x_2[6])$ in Fig. 3b), the algorithm returns to Step 1. Otherwise, the algorithm goes to the next step.
- *Step* 3: Find a cluster from all existing cluster centers by calculating $Si\_j[k] = Di\_j[k] + CDi\_j$, $j = 1, 2, \ldots, R$, then choosing the cluster center with the minimum value:

$$S\min_{i\_j}[k] = D\min_{i\_j}[k] + CD_{i\_j},\ \text{where } j = 1, 2, \ldots, R. \tag{7}$$

  In Eqs. (5) and (6), the minimum distance from any cluster mean to the examples that belong to the cluster is not greater than the threshold D*thr*, though the algorithm does not keep any information of previous examples. However, we find that the formulation only considers the distance between the input data and the cluster mean in Eq. (7). But the special situation (Kwon et al., 1999) shows that the distances between the given point x_i (Hsu, 2003) and both cluster means *Cci*_1 and *Cci*_2 are the same as in Fig. 4. In the aforementioned technique, the cluster C2, which has small dimension distances *CDi*_2, will be selected to expand according to Eq. (7). However this causes a problem in that the cluster numbers increase quickly. To avoid this problem, we state a condition, as follows: If there are two D min $j[10]$ computed in Eq. (6) that $D\min_1[10] = D\min_2[10]$ and $(CD_{1\_1} + CD_{2\_1} > CD_{1\_2} + CD_{2\_2})$

$$Then\ D\min_{1\_1}[10] = D_{1\_1}[10], \tag{8}$$

$$D\min_{2\_1}[10] = D_{2\_1}[10], \tag{9}$$

  where $(D\min_1[10])$ represents the minimum distance between the 10th input data and the mean of the 1st cluster that is calculated by Eq. (6); $(D\min_{2\_1}[10])$ represents dimension distance between the 2nd dimension of the 10th input data and the 2nd dimension mean of the 1st cluster that is calculated by Eq. (7); $D_{2\_1}[10]$ represents dimension distance between the 2nd dimension of the 10th input data and the 2nd dimension mean of the 1st cluster that is calculated by Eq. (5). In Eqs. (8) and (9), we find that when the distances between the input data and both clusters are the same, the formulation will choose the cluster that has the large dimension distance CD_{1_1} and CD_{2_1}.
- *Step* 4: If $S\min_{i\_j}[k]$ in Eq. (7) is greater than D*thr*, the input example $x_i\ [k]$ does not belong to any existing cluster. A new cluster is

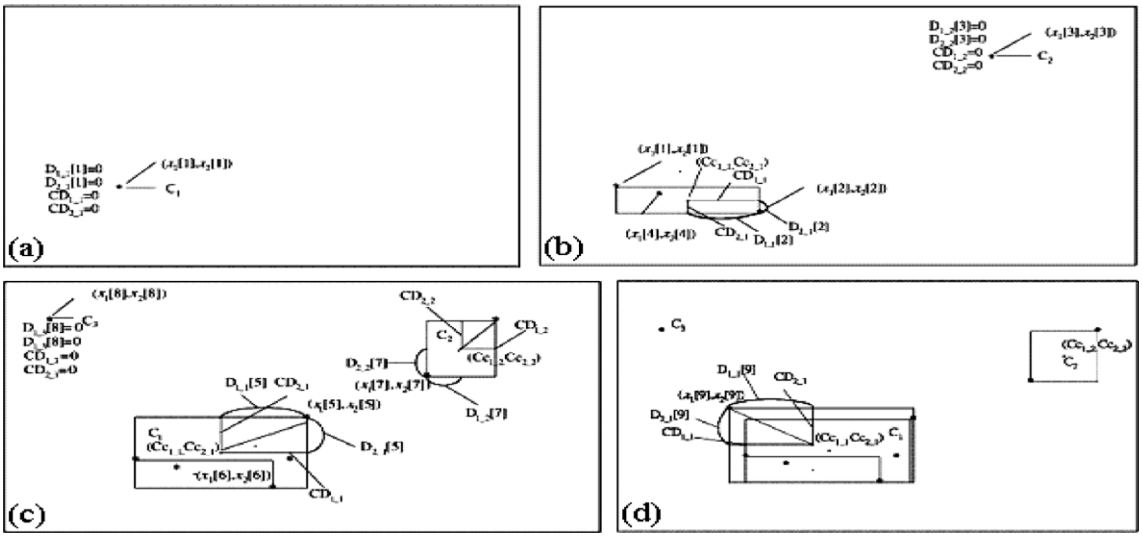


Fig. 3. A brief clustering process using SCA with samples in 2-D space.

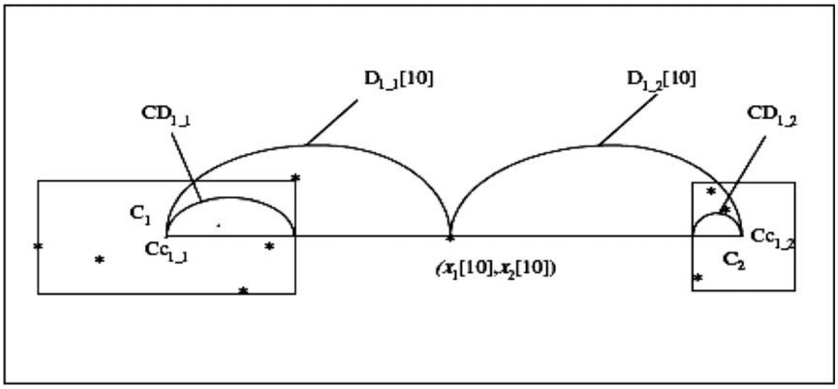


Fig. 4. A special SCA case.

created in the same way as described in Step 0 (the cases of $(x_1[3], x_2[3])$ and $(x_1[8], x_2[8])$ in Fig. 3c), and the algorithm returns to Step 1.

- *Step* 5: If *S* min*i_j* [k] is not greater than D*thr*, the cluster is updated by moving its mean, *Cci_j*, and increasing the value of its dimension distances. The new mean is moved to the point on the line connecting the input data, and the distance from the new mean to the point is equal to its dimension distance (the cases of $(x_1[5], x_2[5])$ and $(x_1[9], x_2[9])$ in Fig. 3d). The details for updating the equations are as follows:

$$\begin{aligned} &\mathit{if}\ CD_{i\_j} < (x_i[k] - Cc_{i\_j} + CD_{i\_j})/2 \\ &\mathit{Then}\ CD_{i\_j} = (x_i[k] - Cc_{i\_j} + CD_{i\_j})/2, \end{aligned} \tag{10}$$

$$Cc_{i\_j} = x_i[k] - CD_{i\_j} \text{ if } Cc_{i\_j} >= x_i[k], \tag{11}$$

$$Cc_{i\_j} = x_i[k] + CD_{i\_j} \text{ if } Cc_{i\_j} < x_i[k], \tag{12}$$

where $k = 1, 2, 3, \ldots, N$ represents the *k*th input, *j* represents the *j*th cluster that has a minimum distance in Eq. (4), *x* represents the input data, and *i* represents the *i*th dimension. After this step is performed, the algorithm returns to Step 1.

After the steps of SCA, we can find some of the more obvious representatives of several groups from cumulative values of each pixel. Then, we get several groups from SCA and into the fuzzy C-mean to analyze. The details of the fuzzy C-mean algorithm are described in the following steps:

The fuzzy C-mean algorithm is based on the minimization of the following criterion function (Eq. (13)) which is the sum of the squared Euclidean distances between an input sample and a cluster center, weighted by the fuzzy membership function.

$$J_{FCM} = \sum_{i} \sum_{k=1}^{C} \mu_k(i)^q \|F(i) - v_k\|^2 \tag{13}$$

In this formula $\mu_k(i,j)$ is the *k*th membership function on the *i*th input sample, and $\sum_{K=1}^{C} \mu_k(i) = 1$, *vk* is the *k*th cluster center, *C* is the amount of clusters, *q* is a parameter of the degree of fuzziness, and $\| \cdot \|$ is the Euclidean distance method. This algorithm iteratively updates the following equations:

$$\mu_k(i) = \left[\sum_{n=1}^{C} \left(\frac{|F(i) - v_k|}{F(i) - v_n} \right)^{2/(q-1)} \right]^{-1}, \quad 1 \leqslant K \leqslant C \tag{14}$$

$$v_k = \frac{\sum_i u_k(i)^q \times F(i)}{\sum_i u_k(i)^q}, \quad 1 \leqslant K \leqslant C \tag{15}$$

The equations are updated until the change of J_{FCM} in Eq. (13) reaches a pre-specified small number, the center locations then become optimal. Once the centers are refined, every input sample is assigned to its nearest center, thus the segmentation is achieved.

Before we use fuzzy C-mean algorithm to segment images, we should assign the c and the initial cluster centers because we want to get values that separate out the light component and dark component. To do so, we only need a small number of clusters. If there are too many regions, it will reduce the performance and the subject region may not complete.

After the steps of fuzzy C-mean, we can get two values that separate out the light component and dark component from the entire image and intensity profile of roadway part block (see Fig. 5).

After, we chose distance estimate method to calculate COA and the two values (light and dark) from the entire image. We chose a smaller value knowing that the COA deflection is a smaller value. Therefore, through the above-mentioned method, we can know the brightness changes of the current image. We also chose to analyze the distance between light and dark of variation through intensity profile of roadway part block.

Next, a fuzzy rule-based system was constructed with the fuzzy "***IF X AND Y THEN Z***" rule. In this study, two basic parameters were taken as the premises of the fuzzy rules: the distance estimate (d) and the light and dark variation (v). After we tested some images, we found that the performance and the completeness of bright compensation are satisfying when the two parameter fuzzy rules

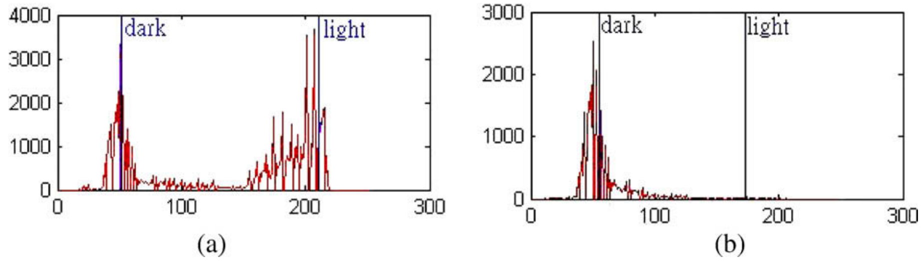
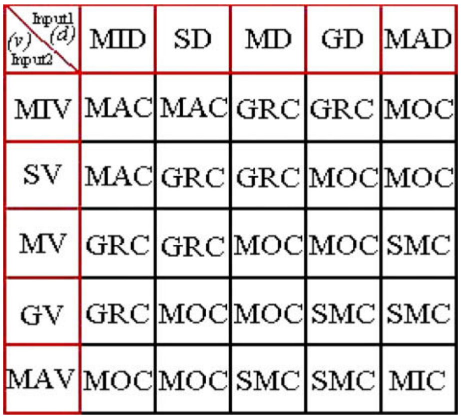


Fig. 5. (a) The two values that separate out the light and dark component from entire image and (b) the two values that separate out the light and dark component from intensity profile of roadway part block.

are set to 5. The distance estimate d was divided into five membership functions: **MAD** represents maximum distance, **GD** represents greater distance, **MD** represents middle distance, **SD** represents smaller distance, and **MID** represents minimum distance. The light-dark variation v is divided into five membership functions: **MAV** represents maximum variation, **GV** represents greater variation, **MV** represents middle variation, **SV** represents smaller variation, and **MIV** represents minimum variation. There were five gradation lighting compensation outputs: **MAC** represents maximum compensation, **GRC** represents greater compensation, **MOC** represents moderate compensation, **SMC** represents smaller compensation, and **MIC** represents minimum compensation, controlled by the consequential outputs from the fuzzy rules. We therefore established 25 fuzzy logic rules as listed below and shown in Fig. 6.



Input1 (d) (v) Input2	MID	SD	MD	GD	MAD
MIV	MAC	MAC	GRC	GRC	MOC
SV	MAC	GRC	GRC	MOC	MOC
MV	GRC	GRC	MOC	MOC	SMC
GV	GRC	MOC	MOC	SMC	SMC
MAV	MOC	MOC	SMC	SMC	MIC

Fig. 6. Fuzzy rule table for brightness compensation control units (d: distance; v: light-dark variation).

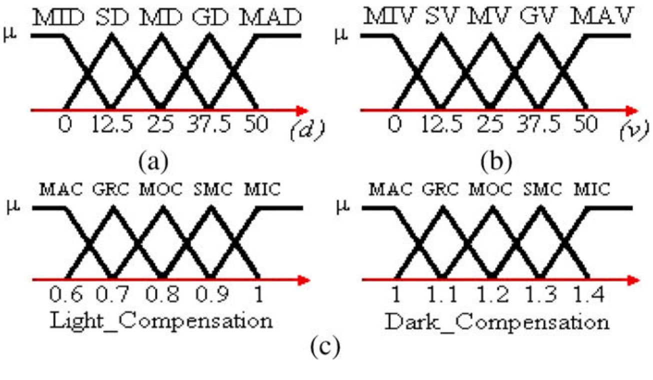


Fig. 7. Fuzzy logic membership functions.

The membership functions for the brightness compensation are shown in Fig. 3.8, where (a) is the membership functions of distance estimate, (b) is the membership functions of light-dark variation, and (c) is the membership functions of output operations. For the distance d, as shown in Fig. 7a, when $d < 0$, we consider it is ‘‘very minimum distance”, and ‘‘very maximum distance” when $d > 50$. In Fig. 7b, when the light-dark variation v is larger than 50, we say it is ‘‘very maximum variation”, and when the value of v is smaller than 0, we say it is ‘‘very minimum variation”. Fig. 7c shows the consideration of comfortable bright compensation for all kinds of the weather.

In this study, the compensation scale is set to between 0.6 and 1.4. The μ represents the degree of membership of different members. It describes the membership of both the premise term and the consequent term of the fuzzy rules.

Next, we got the output from consequent operation and substitution ‘‘a” of the parameter in Eq. (16). We used through Eq. (16) to calculate brightness compensation of each pixel in the image. Fig. 8a is parameter ‘‘a” of light compensation curve, where as Fig. 8b is parameter ‘‘a” of dark compensation curve.

$$compensation\_pixel(i,j) = \left(\frac{pixel(i,j)}{255}\right)^{a} \times 255 \qquad (16)$$

After the above method, we can get better images information and enhance the accuracy of follow-up procedure in Fig. 9. Fig. 9a is the original gray image, where as Fig. 9b is the passed through bright compensation gray image. We can see (see Fig. 9b) of the more obvious advantages of that approach in the areas circled in red.

3.2. Canny edge detection

Edges characterize boundaries and therefore a problem of fundamental importance in image processing. Edge detecting an image significantly reduces the amount of data and filters out useless out information, while preserving the important structural properties in an image.

The Canny edge detection algorithm (Gonzalez, Woods, & Eddins, 2003) is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, ‘‘A Computational Approach to Edge Detection”. In (Gonzalez et al., 2003), Canny followed a list of criteria to improve current methods of edge detection. The first and most obvious criterion is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not

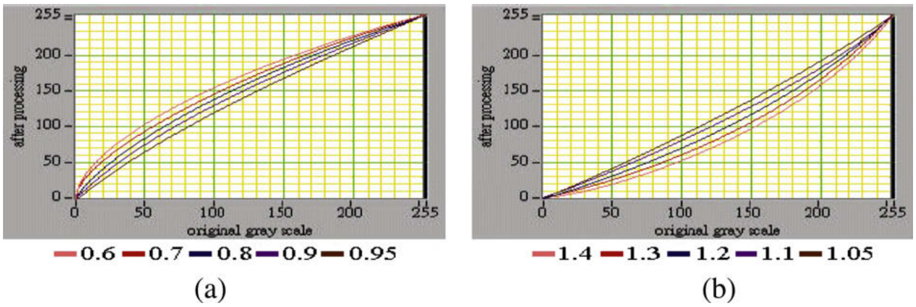


Fig. 8. The brightness compensation curve.

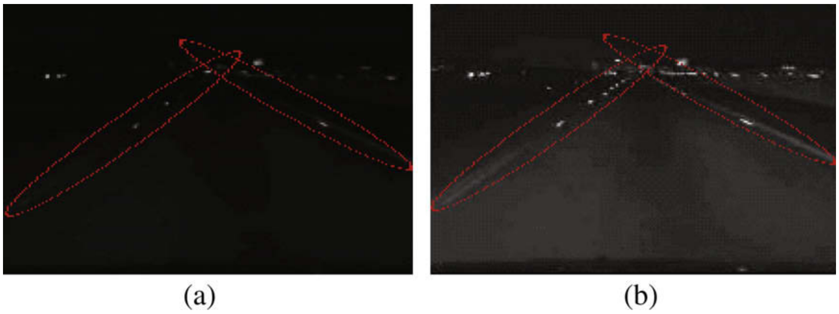


Fig. 9. (a) The original gray image and (b) the passed through brightness compensation gray image.

substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the Canny detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is path from this pixel to a pixel with a gradient above T2. Applying the above methods, the detection results are shown in Fig. 10, Fig. 10a shows the edge detection without brightness compensation. Fig. 10b shows the edge detection with brightness compensation and according to the figure red circle regions shows that brightness compensation can be enhanced the lane information.

3.3. Fan scanning detection

We will be scanning the images from a bottom-up, the middle to both sides, where the first encounter of an edge pixel in each row is saved and deleted all the other point. The process is continued as edge points are linked to form line segments in the image. These line segments will be combined into lane line through the following steps.

- *Step* 1: Calculate coordinate difference between the current edge pixel with the previous edge pixel, If $\Delta y < 3$ and $0 \leqq \Delta x < 6$, then the point joins the previous edge pixel in the list. If not, then the following principles to decide.
- *Step* 2: Calculate coordinate difference between the current edge pixel with the preceding every list of end point. If $\Delta y \leqq 3$ and $0 \leqq \Delta x \leqq 6$, then, the point accedes to the list of the homology. If not, then this edge pixel builds into a new list.
- *Step* 3: Based on the judgment of the two prior principles, $\Delta x = 0$ situation is not permitted to continuous appear two times or more times, to avoid finding out the vertical direction of line segment.

Next, we have to get the line segments to link together to form the lane line. We were respectively the list of the left-right sides proceeding sort, and based on the number size of the edge point. Then, we took out the longest list's the start point and end point from left to right sides, to estimate a straight line. Next, we calculate the left–right side straight lines the intersection location; the point is the preliminary estimates point to the image vanishing point. It is calculated using Eqs. (17) and (18), where and representing the image the left (or right) in the longest list of the start point and end point coordinates. In addition, *a*, *b*, *c* of straight line equations $ax + by + c = 0$ three factors, and the subscript *L* and *R* on behalf of the left or the right of the straights coefficient make up the equation.

$$\begin{aligned} a &= y_1 - y_2 \\ b &= x_2 - x_1 \\ c &= x_1y_2 - x_2y_1 \end{aligned} \tag{17}$$

$$vy = \frac{a_Lc_R - a_Rc_L}{a_Rb_L - a_Lb_R} \tag{18}$$

In the formula (18) *vy* is the intersection *y* coordinate for the in two straight lines as well as the *y* coordinates of the vanishing point. Therefore, edge points of the *vy* and upward do not have to be dealt with, so that we will be able to prevent vehicles and road scenery generation of edge point to cause an incorrect detection of the lane. Then, we have to consider the left and right sides of the link list respectively, from the second long list to the shortest list and check whether to merger with the longest list. If so, we will combine the two lists; if not, we will continue to check the next list. The principles of the merger as follow:

- *Step* 1: We first take out the longest list the start point and end point coordinates (the *A* line segment in Fig. 11).

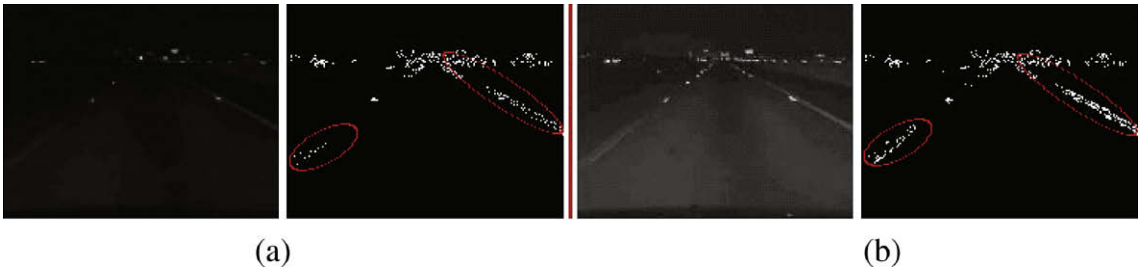


Fig. 10. (a) The edge detection without brightness compensation (b) the edge detection with brightness compensation.

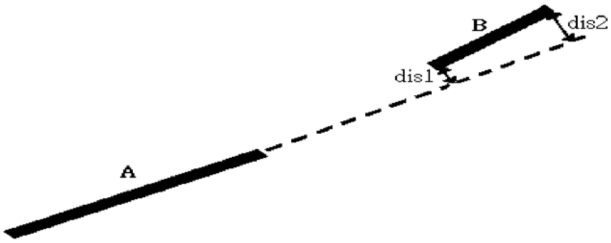


Fig. 11. Sketch of the merger line segment.

Fig. 12. The result of the link in the same lane line segment.

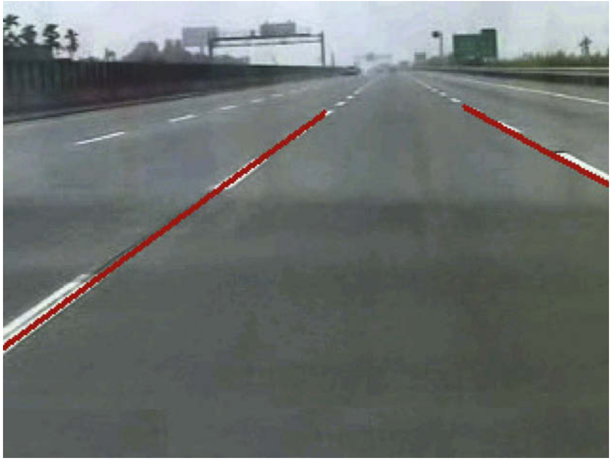


Fig. 13. The result of the depicted in actual image.

- *Step* 2: Then, we use these two points to estimate a straight line (the dotted line in Fig. 11) and it is factor calculation formula as shown in (17).
- *Step* 3: Then, we take out the start point and end point coordinates from the list of the merger (the *B* line segment in Fig. 11).
- *Step* 4: We calculate distance from to two points, dis1 and dis2, as follow:

$$dis = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}} \tag{19}$$

- *Step* 5: If $dis1 < th1, dis2 < th2$, then we will combine the two lists and to get a new list.

When the road is not straight, the two may not be in the same segment of the straight line (Fig. 11 of the *A, B* line segment). Therefore (*th1, th2*) will be a distance of two line segments to calculate thresholds. That is the farther the distance, the greater the threshold set. So the threshold is defined as follows:

$$threshold = \frac{ya - yb}{4} + 2 \tag{20}$$

In the formula, *ya* want to merge the list to *y* coordinate of the start point or end point. If the new list at the top of the longest list, *yb* is *y* coordinates from the end of the longest list. If the new list at the under of the longest list, *yb* is *y* coordinates from the start of the longest list. If the new list in the longest at the middle of the list, the threshold set 2. The result is shown in Fig. 12.

After, the line segments are merged using start point and end point to depict the actual image (see Fig. 13).

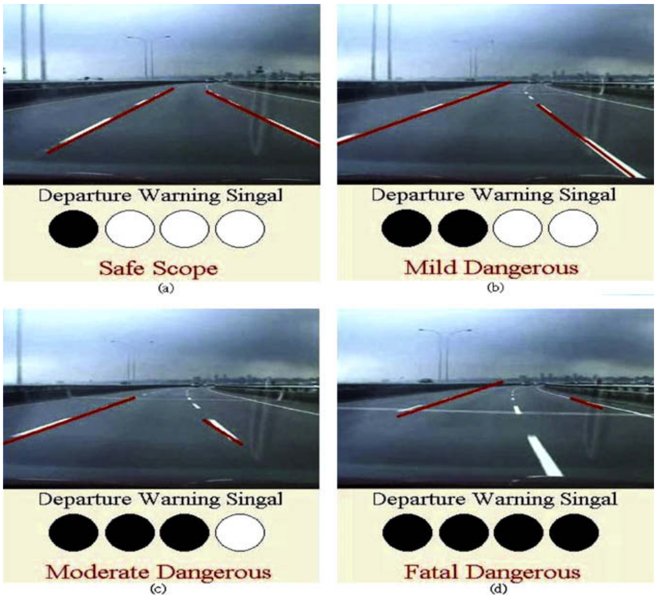


Fig. 15. Results of the various warning signal display.

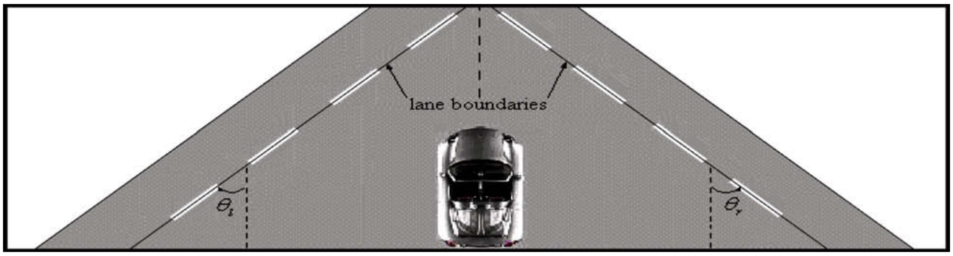


Fig. 14. Orientation of lane boundaries.

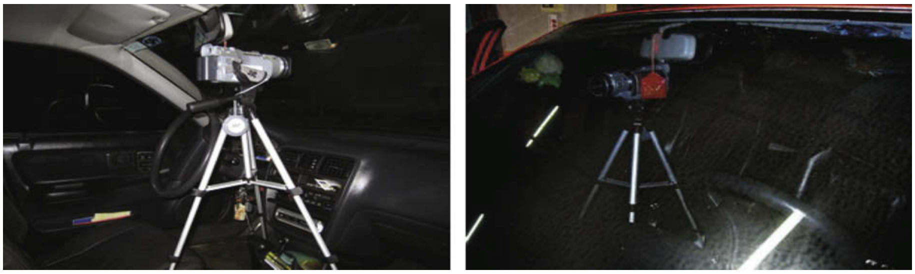


Fig. 16. The experimental architecture.

4. Lane departure warning

We use measurement, namely, the angles of both sides of lane lines to identify if the hazards to road driving arise. Therefore, how to evaluate the angles will be shown in the follows. If the vehicle is traveling in a straight portion of the road and stays at the center of the lane, we should expect symmetry (for the near vision field) in the orientations of left and right lane boundaries($\theta_l + \theta_r = 0$), as depicted in Fig. 14. If the vehicle drifts to its left, both θ_l and θ_r increase. If the vehicle drifts to its right, both θ_l and θ_r decrease. In any case, the value $|\theta_l + \theta_r|$ gets away from zero. Thus, a simple and efficient measure for trajectory deviation is given by:

$$\beta = |\theta_l + \theta_r| \tag{21}$$

If β gets sufficiently large, the vehicle is leaving the center of the lane. In this work, β is compared to a threshold T1, and a lane departure warning is issued if $\beta > \text{T1}$. Experimental results indicate that $T1 = 15^\circ$ is a good choice.

In the previous chapter 3, we have derived explicit models for left and right lane boundaries, that are denoted by $f_l(x)$ and $f_r(x)$, respectively. Such models can be used to determine orientations $\theta_l(x)$ and $\theta_r(x)$:

$$\theta_l(x) = \tan^{-1}\left(f_l'(x)\right), \quad \theta_r(x) = \tan^{-1}\left(f_r'(x)\right) \tag{22}$$

Eq. (22) provides orientations $\theta_l(x)$ and $\theta_r(x)$ at any position x. However, to compute the symmetry measure β, we need to determine such orientations in the near field (to obtain the vehicle's current orientation). In fact, $\theta_l(x) = \theta_l$ and $\theta_r(x) = \theta_r$ are actually constant values within the near field, because $f_l(x)$ and $f_r(x)$ are both linear functions for any $x \geqslant x_m$.

Table 1
The specification of platform information.

CPU	Intel Pentium 4 3.2 GHz
Memory	1 GB DDR400 RAM
Compiler	Borland C++ Builder 6.0
OS	Microsoft Windows XP
Resolution	320 × 240
Frame rate	30 FPS

In order to reduce the influence of noise, temporal filtering is applied to estimate θ_l and θ_r, by averaging orientations in consecutive frames. If θ_l^k and θ_r^k denote orientations in the kth frame of the video sequence, then orientations in the current frame n are given by:

$$\theta_l^n = \sum_{k=0}^{9} \theta_l^{n-k} \text{ and } \theta_r^n = \sum_{k=0}^{9} \theta_r^{n-k} \tag{23}$$

In general, lane departures can be classified as wanted or unwanted. In the first case, the driver makes a voluntary lane change (to overtake a car, for example), and turns on the blinker to indicate his intent. The second case corresponds to involuntary lane changes that usually occur when the driver falls asleep or is not paying attention to the road. These cases are treated differently by our algorithm.

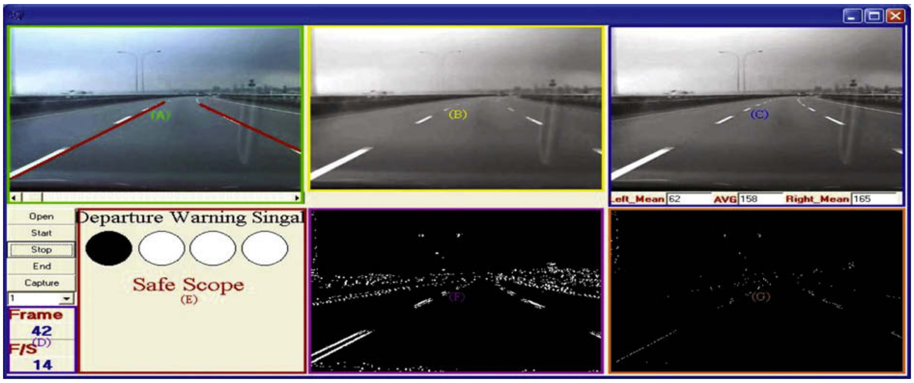
Based on the above formula (21) and (23), we have to calculate the extent of deflect and give different levels warning signal. The judge formula 24, 25 and warning signal display as follows (see Fig. 15):

- *Step* 1:

$$\textit{if}\ \beta > T1\ \textit{then departure\_number} = \textit{departure\_number} + 1 \tag{24}$$

- *Step* 2:

$$\begin{aligned}
&\textit{if departure\_number} < 3\\
&\quad \textit{then}(\textit{Safe Scope})\\
&\textit{Else if departure\_number} \geqslant 3 \textit{ and departure\_number} < 5\\
&\quad \textit{then}(\textit{Mild Dangerous})\\
&\textit{Else if departure\_number} \geqslant 5 \textit{ and departure\_number} < 8\\
&\quad \textit{then}(\textit{Moderate Dangerous})\\
&\textit{Else if departure\_number} \geqslant 8\\
&\quad \textit{then}(\textit{Fatal Dangerous})
\end{aligned} \tag{25}$$



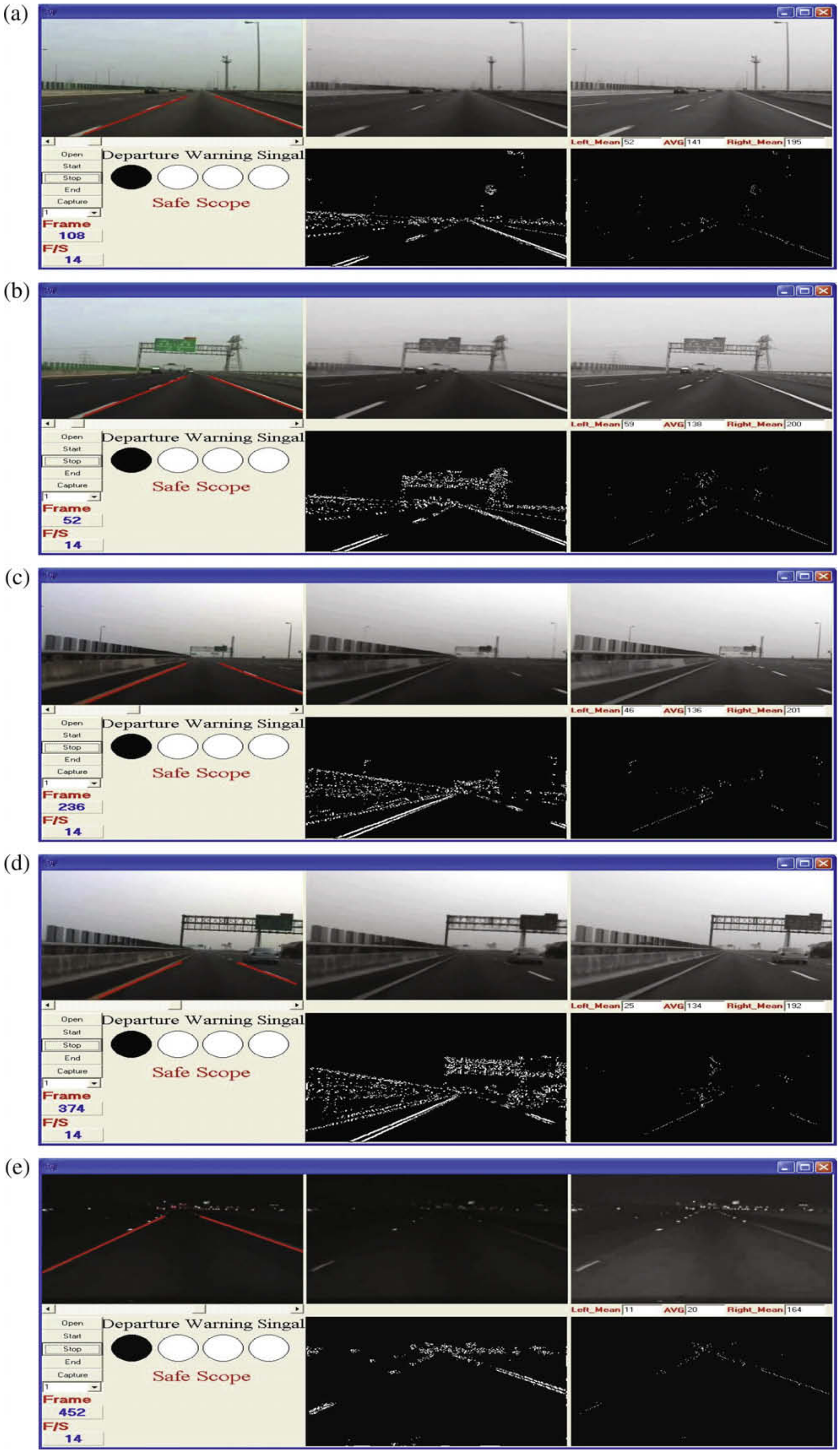
Left_Mean 62 AVG 158 Right_Mean 165

1

Frame 42

F/S 14

Fig. 17. The programming interface in the PC platform.



(a)
Left_Mean 52 AVG 141 Right_Mean 195
Frame 108
F/S 14

(b)
Left_Mean 59 AVG 138 Right_Mean 200
Frame 52
F/S 14

(c)
Left_Mean 46 AVG 136 Right_Mean 201
Frame 236
F/S 14

(d)
Left_Mean 25 AVG 134 Right_Mean 192
Frame 374
F/S 14

(e)
Left_Mean 11 AVG 20 Right_Mean 164
Frame 452
F/S 14

Fig. 18. The results of lane detection.

5. Experimental results

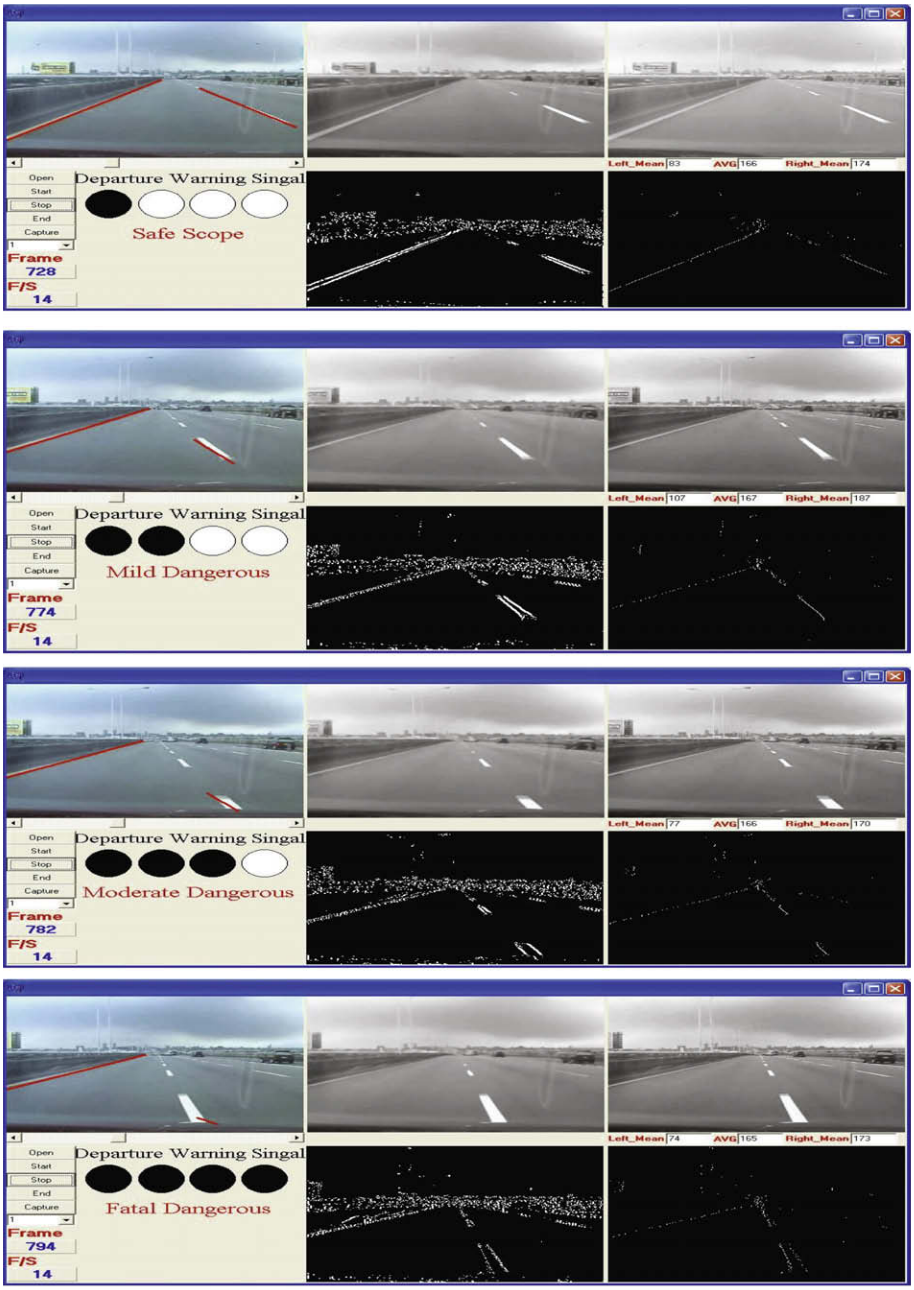
5.1. Experimental setup

Fig. 16 shows that a JVC GR-DV4000U camera is mounted under the rear-view mirror on the side of the vehicle to acquire view image sequences and specification of platform information in Table 1.

Fig. 17 shows the realistic programming interface in the PC platform. Block (A) contains the input frame which is added the approximating straight lane boundary detection, red line, by our methods, as explained in Chapter 3. Block (B) contains the gray image by RGB-to-YIQ, as explained in Section 3.1. Block (C) contains the brightness compensation by fuzzy rule, as explained in Section 3.1. Block (D) show the output frame rate which responds to the systematic performance. The warning alarm with different colors of lane of LDW is contained by block (E). Block (F) displayed the edge detection by Canny algorithm, as explained in Section 3.2. At last, Block (G) display to retain more meaningful information by Fan Detection, as explained in Section 3.3.

5.2. Explanation of experimental conditions

The driving environment is focused on highway with different light conditions. The image sequences captured by the camera are tested. At the same time, in order to observe and prove our methods can maintain robust performance with tolerate the light variation, we select the video segments with five different periods, morning, noon, afternoon, nightfall, and night of one day for experiment in the next section.



Departure Warning Singal
● ○ ○ ○
Safe Scope
Frame 728
F/S 14
Left_Mean 83 AVG 166 Right_Mean 174

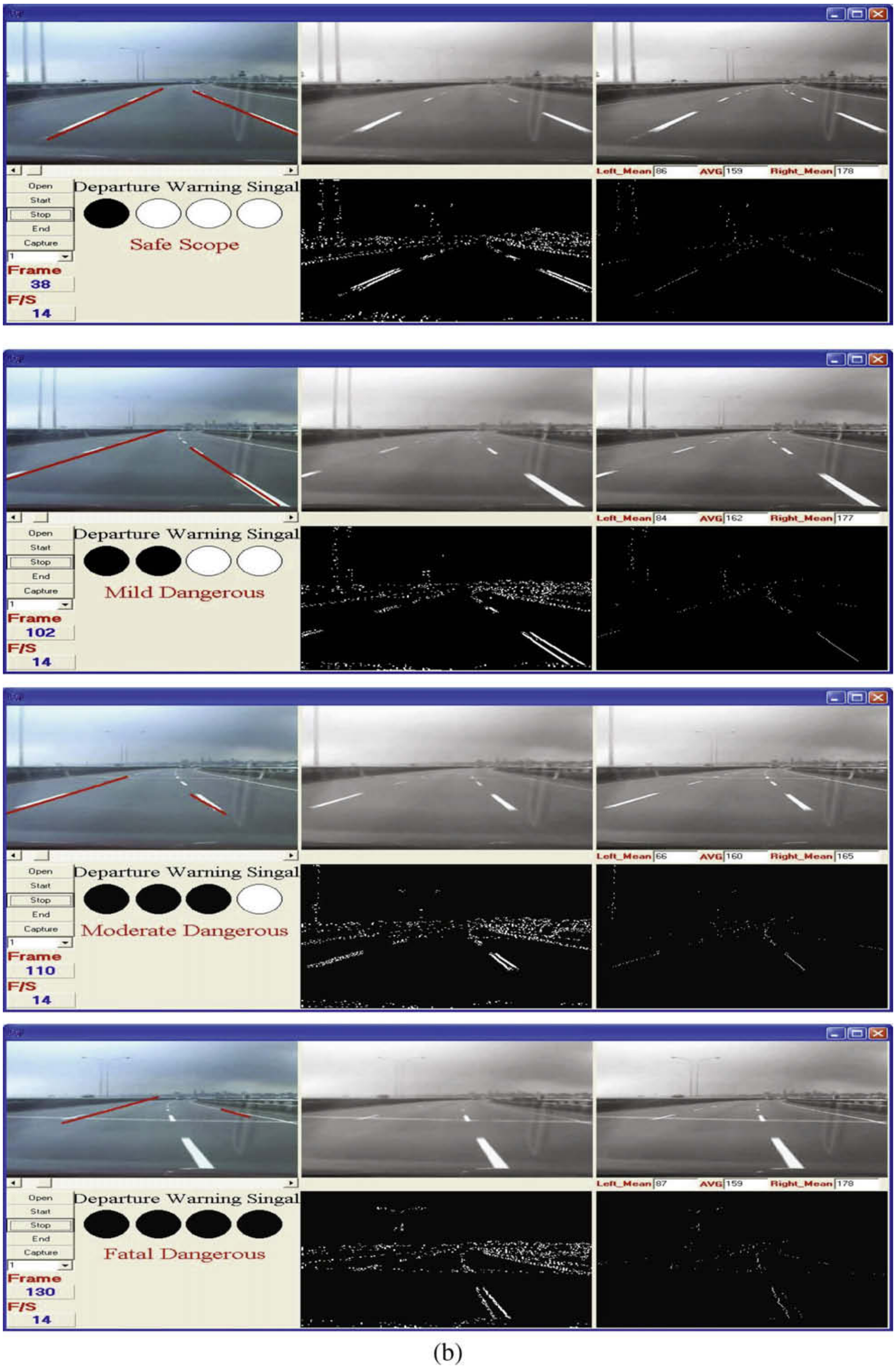
Departure Warning Singal
● ● ○ ○
Mild Dangerous
Frame 774
F/S 14
Left_Mean 107 AVG 167 Right_Mean 187

Departure Warning Singal
● ● ● ○
Moderate Dangerous
Frame 782
F/S 14
Left_Mean 77 AVG 166 Right_Mean 170

Departure Warning Singal
● ● ● ●
Fatal Dangerous
Frame 794
F/S 14
Left_Mean 74 AVG 165 Right_Mean 173

(a)

Fig. 19. The Results of lane departure warning.



Left_Mean 86 AVG 159 Right_Mean 178
Departure Warning Singal
○ ○ ○ ●
Safe Scope
Frame 38
F/S 14

Left_Mean 84 AVG 162 Right_Mean 177
Departure Warning Singal
● ● ○ ○
Mild Dangerous
Frame 102
F/S 14

Left_Mean 66 AVG 160 Right_Mean 165
Departure Warning Singal
● ● ● ○
Moderate Dangerous
Frame 110
F/S 14

Left_Mean 87 AVG 159 Right_Mean 178
Departure Warning Singal
● ● ● ●
Fatal Dangerous
Frame 130
F/S 14

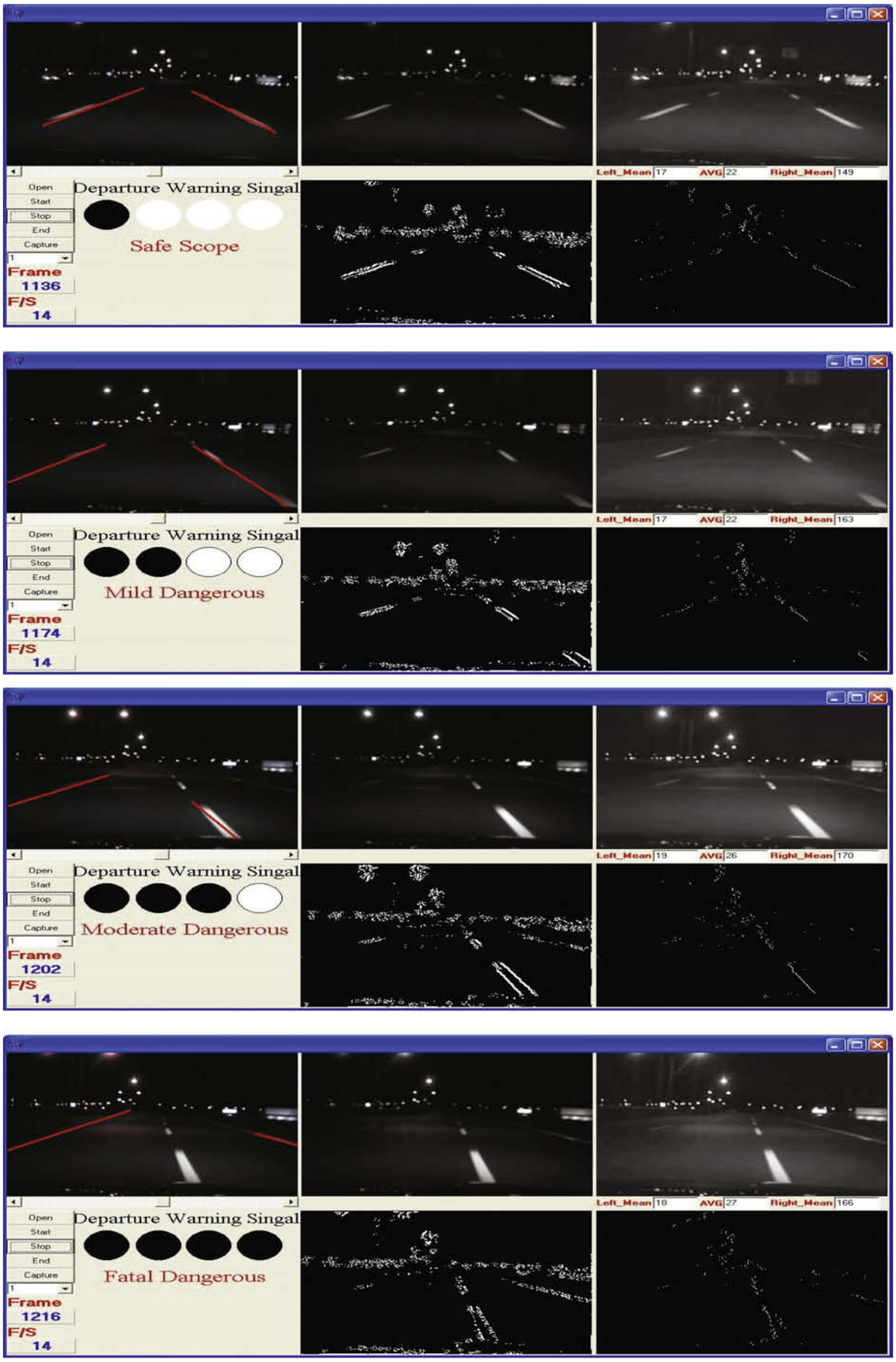
Fig. 19 (*continued*)

In Fig. 18, the testing environment considers the property with respect to the conditions simultaneously. The detection results of morning (a), noon (b), afternoon (c), nightfall (d) and night (e) are processed by the same programming setting.

If the lane boundary is locked precisely by the lane detection mechanism, the lane departing maneuver can be calculated. Through, we measured the angles from our methods and we mete out different warning signal by deviation of different degrees. The departure detection results of morning (a), afternoon (b) and night (c) are processed by the same programming setting in Fig. 19.

6. Conclusions

In the research, we proposed a new lane detection and lane departure warning system based on fuzzy rules to analyze and it can be applied to different weather periods. Initially, a combination of the SCA, fuzzy C-mean and fuzzy rule is used to detect brightness changes in the video scenes and enhance definite information. Next, canny edge detection algorithm and fan detection are used to lane boundaries in the compensation subsequent images. Finally, orientation of the vehicle with respect to both lane boundaries is



Departure Warning Singal
● ○ ○ ○
Safe Scope
Frame 1136
F/S 14
Left_Mean 17 AVG 22 Right_Mean 149

Departure Warning Singal
● ● ○ ○
Mild Dangerous
Frame 1174
F/S 14
Left_Mean 17 AVG 22 Right_Mean 163

Departure Warning Singal
● ● ● ○
Moderate Dangerous
Frame 1202
F/S 14
Left_Mean 19 AVG 26 Right_Mean 170

Departure Warning Singal
● ● ● ●
Fatal Dangerous
Frame 1216
F/S 14
Left_Mean 18 AVG 27 Right_Mean 166

(c)

Fig. 19 (*continued*)

computed, and a symmetry measure is used to detect in lane departure warning system.

Experimental results indicate that the proposed model provides an accurate fit to lane boundaries, and can be used to obtain robust information about their orientation. Also, these orientations are used to produce a symmetry measure that correctly indicates tendencies of lane departure in advance (thus, providing enough time for the driver to correct his/her trajectory). Further work will concentrate on extending the proposed model to estimate orientation of the vehicle with respect to both lane boundaries in world coordinates. This would enable active safety (the car could actually take over the steering wheel to prevent an accident) and autonomous driving.

References

Abdel-Aziz, Y. I. & Karara, H. M. (1971). Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proceedings of the symposium on close-range photogrammetry* (pp. 1–18). Falls Church, VA: American Society of Photogrammetry.

Apostoloff, N., & Zelinsky, A. (2003). Robust based lane tracking using multiple cues and particle filtering. In *Proceedings of IEEE intelligent vehicles symposium*, pp. 558–563.

Bas, E. K., & Crisman, J. D. (1997). An easy to install camera calibration for traffic monitoring. In *IEEE conference on intelligent transportation system*, November 9–12, 1997.

Chen, M., Jochem, T., & Pomerleau, D. (1995). AURORA: A vision-based roadway departure warning system. In *Proceedings of the IEEE International Conference on Robotics and Automation* (Vol. 1, pp. 43–248).

Echingo, T. (1990). A camera calibration technique using three sets of parallel lines. *Machine Vision and Applications, 3*, 159–167.

Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2003). *Digital image processing*. Prentice-Hall, Inc.

Hsu, C. S. (2003). The decision strategies of irregular vehicle behavior warning system, M.S. thesis, Graduate Institute of Civil Engineering, National Taiwan University, Taipei, Taiwan.

Hsu, P. L., Cheng, H. Y., Tsuei, B. Y., & Huang, W. J. (2002). The adaptive lane-departure warning system. In *Proceedings of the 41st SICE annual conference* (Vol. 5, pp. 2867–2872).

John, Y., & Langari, R. (1999). *Fuzzy logic: Intelligence control and information*. Prentice-Hall, Inc.

Jung, C. R., & Kelber, C. R. (2004). A lane departure warning system based on a linear-parabolic lane model. In *IEEE intelligent vehicles symposium* (pp. 891–895), June 14–17.

Jung, C. R., Kelber, & C. R. (2005). A lane departure warning system using lateral offset with uncalibrated camera. *Proceedings of the eighth international IEEE conference on intelligent transportation systems* (pp. 13–16).

Kwon, W., Lee, J. W., Shin, D., Roh, K., Kim, D. Y., & Lee, S. (1999). Experiments on decision making strategies for a lane departure warning system. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 2596–2601).

LeBlanc, D. J., Johnson, G. E., Venhovens, P. J., Gerber, G., Sonia, R. D., Ervin, R. D., et al. (1996). CAPC: A road departure prevention system. *IEEE Control Systems Magazine, 16*, 61–71.

Lee, J. W. (2002). A machine vision system for lane-departure detection. *Computer Vision and Image Understanding, 86*(1), 52–78.

Lee, J. W., Kee, C. D., Yi, U. K. (2003). A new approach for lane departure identification. In *Proceedings of IEEE intelligent vehicles symposium* (pp. 100–105).

Lin, C. J., & Xu, Y. J. (2006). A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications. *Fuzzy Sets and Systems, 157*(8), 1036–1056.

McCall, J. C., & Trivedi, M. M. (2004). An integrated robust approach to lane marking detection and lane tracking. In *Proceedings of IEEE intelligent vehicles symposium* (pp. 533–537). Italy: Parma.

Risack, R., Mohler, N., & Enkelmann, W. (2000). A video-based lane keeping assistant. In *Proceedings of IEEE intelligent vehicles symposium* (pp. 356–361).

Thomas, B. (2000). Measurement of distance and height in images based on easy attainable calibration parameters. In *Proceedings of the IEEE intelligent vehicles symposium* (pp. 314–319).

Wang, L. L., & Tsai, W. (1991). Camera calibration by vanishing lines for 3-D computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(4), 370–376.