

## Research Article

# A Fuzzy Rules-Based Driver Assistance System

Jong-Chih Chien,<sup>1</sup> Jiann-Der Lee,<sup>2</sup> and Li-Chang Liu<sup>2</sup>

<sup>1</sup>Department of Information Management, Kainan University, No. 1 Kainan Road, Luzhu Shiang, Taoyuan 33857, Taiwan

<sup>2</sup>Department of Electrical Engineering, Chang Gung University, 259 Wen-Hwa 1st Road, Taoyuan 333, Taiwan

Correspondence should be addressed to Jiann-Der Lee; [jdlee@mail.cgu.edu.tw](mailto:jdlee@mail.cgu.edu.tw)

Received 5 March 2014; Accepted 31 October 2014

Academic Editor: Valentina Emilia Balas

Copyright © 2015 Jong-Chih Chien 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.

A vision-based driver assistance system using fuzzy rules to determine whether warnings are necessary is presented. This system is comprised of four cameras, one of which focuses on the driver for estimating the driver's viewing angle, another focuses on the road ahead for the detection of road condition ahead, and the last two cameras are on both sides of the vehicle, facing backward, for the purpose of determining whether neighboring lanes are occupied by vehicles hidden in the blind-spots. The system uses fuzzy-rules for the analysis of interactions between the driver's gaze, whether there are vehicles ahead, and in the neighboring lanes to determine whether the current driving condition should be of concern to the driver and issues one of three levels of warnings, from safe to dangerous.

## 1. Introduction

As the number of traffic accident rises each year, so does the need for a sophisticated driver assistance system become more apparent. This paper proposes a vision-based driver assistant system which would issue alerts when the vehicle in unsafe situations. There were other studies into smart driver assistants [1–5], but most of them do not know how to deal with the correlation between the driver's attention and the vehicle's environment. For example, Trivedi et al. [4] and his team proposed the LiLo system which would utilize multiple types of cameras for tracking driver's intents and car movements, but no detail was provided on how and if they were to interact. Our proposed driver's safety system simultaneously evaluates the car's motions as well as the driver's visual focus in order to avoid accidents caused by momentary lapses of attention.

The information from the forward-facing camera is used to locate and classify the lane boundaries, which are then used to determine the deviation of the vehicle from the lane center as well as the distance to a vehicle ahead. The data from the backward-facing cameras are used to determine direction of the driver's gaze and to determine the existence of vehicles in the neighboring lanes. These pieces of extracted

information are then fed to the fuzzy analysis modules to determine whether to warn the driver. The camera setup for our system is shown in Figure 1.

The analysis portion of our driver safety system is composed of three fuzzy-rules-based modules, and a crisp-rules-based module each of which would assess its inputs and issue its own evaluation of the driver safety level, and the most severe warning amongst them would be then issued by the system. The safety level is divided into three tiers: "safe," "risky," and "dangerous." The analysis modules are fuzzy rules-based driving safety analyses in the presence of obstacles, fuzzy-rules-based driving safety analysis in obstacle-less driving situations, the fuzzy rules-based driving safety analysis for the blind-spots, and a crisp-rule based analysis module that determines whether the car's movement is legal; an obstacle is defined as a vehicle in front.

The organization of this paper is as follows: the driver safety system's overview will be presented in Section 2, how each piece of information is extracted will be discussed in Section 3, how the information is analyzed will be presented in Section 4, a lab simulation of the entire system using the parameters from real-world driving situations will be presented in Section 5, the results of our experiments and

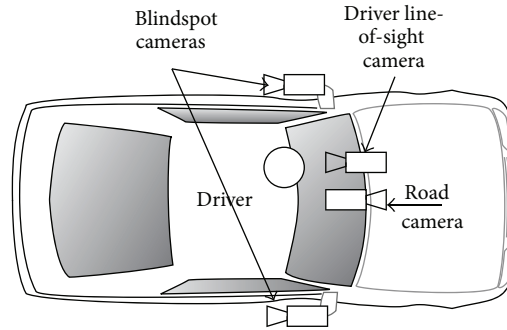


FIGURE 1: Positions of the cameras for our driver safety system.

comparison of the methods we used in our system with other methods will be presented in Section 6, and conclusions follow in the final section.

## 2. System Description

Useful pieces of information are extracted from raw data from both the forward-facing and backward-facing cameras. These pieces of information include the locations and types of lane boundaries, the deviation of the vehicle from the center of the lane, and the deviation of alignment between the driver's line-of-sight and the center of the driving lane, whether a vehicle exists ahead and whether vehicles exist in the neighboring lanes. In the case where a vehicle ahead is detected, information such as deviation of driver's line-of-sight from the obstacle and the distance to the obstacle will also be calculated or estimated. Each piece of data is fuzzified before being sent to be analyzed in the analysis modules.

The system flow chart is shown in Figure 2. It shows that instead using the distance-to-vehicle as an input value directly into the analysis modules; it was first divided by the safety distance so as to express it as a percentage of the safety distance or a normalized distance value. The safety distance is the minimum braking distance between two moving vehicles. Because the safety distance is dependent mostly on the speed of the vehicle, so we use the general rule of thumb of one vehicle length per every 8 km/h to determine the minimum safety distance.

As shown in Figure 2, from right to left, the fuzzy-rules-based safety analysis for obstacle-less driving module takes as input: the driver's sight deviation from lane center, vehicle's deviation from lane center, and speed. The fuzzy-rules-based safety analysis in the presence of obstacle(s) takes as input: the deviation of the driver's sight from obstacle, vehicle's deviation from lane center, and the normalized distance to obstacle. The legal-maneuver module uses the types of lane boundaries and the vehicle deviation as input. Lastly, the fuzzy-rules-based analysis for blind-spot detection takes as input: the presence or absence of vehicle in blind-spots, the driver's gaze angle, and the vehicle's deviation from lane center. The following section will discuss how these raw data turn into useful information for the analysis modules.

## 3. Information Extraction

**3.1. Driver's Viewing Angle.** As illustrated by the flow chart in Figure 3, the driver's viewing angle extraction stage uses as its input the driver's image; it then detects the driver's face by using the Haar-like feature-based Adaboost classifier mentioned in [6]. Once the driver's face is detected, it is continuously tracked using a particle-filtering-based tracker [7]. The particle-filter tracking method was chosen because its robustness in the presence of occlusions is well known and has shown to have better performance [8] in head tracking applications than the traditional Kalman filtering. However, tracking face alone is insufficient in determining the driver's facial orientation, the position of the driver's eyes relative to the position of the nose also helps to determine the face's orientation. For determining the eyes' location, another Haar-feature like Adaboost classifier was trained to locate the eyes within the facial region being tracked, and the position of the pupils within their sockets is used to determine the glancing angle which is used to adjust the driver's facial orientation to yield the final line-of-sight angle. The flow chart of the particle-filtering-based tracking process is illustrated in Figure 4.

In our system, we used the color histograms as the basis for our particle filter tracker for tracking driver's head. From the camera image, the pixels enclosing the driver's head form the reference window and its color histogram is computed. The target windows are formed by the possible locations where the driver's head would be next. The intersection values of the color histograms of the reference window with the target windows are calculated and used to weigh the samples, by giving the target windows with larger intersection values the larger weight. Following the steps illustrated in Figure 4 until a result can be estimated. The next stage after tracking is the determination of the driver's line-of-sight.

The driver's line-of-sight is defined as the facial-orientation angle adjusted by the angle of the pupils off the centers the eye sockets. In order to determine the facial orientation, the positions of the eyes and the nose are to be used. But while the locations of the face and eyes can be found using Haar-features-based Adaboosted classifiers, the location of the nose would have to be estimated. This process of estimation is done by putting the locations of the eyes

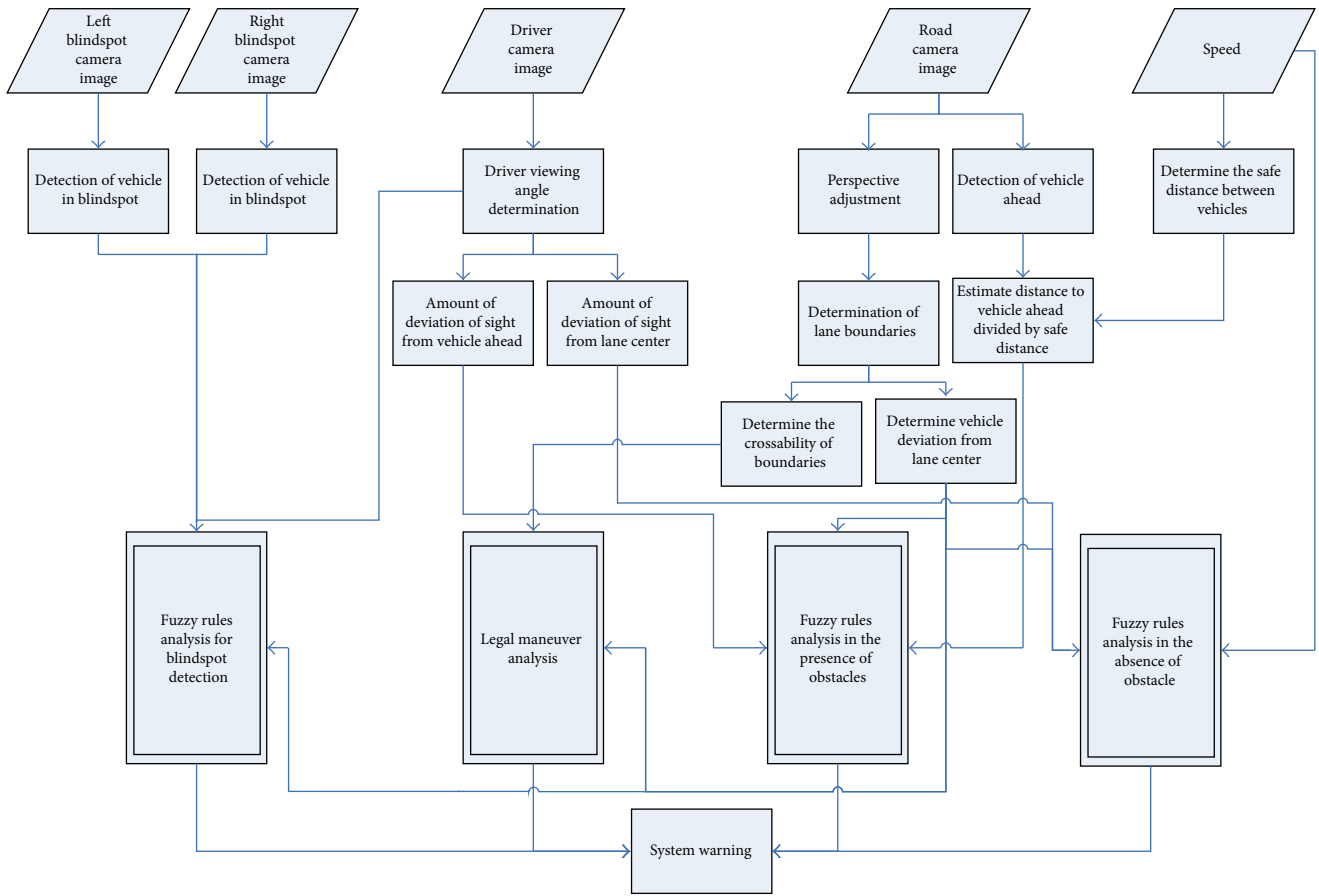


FIGURE 2: System flowchart for the driver assistant.

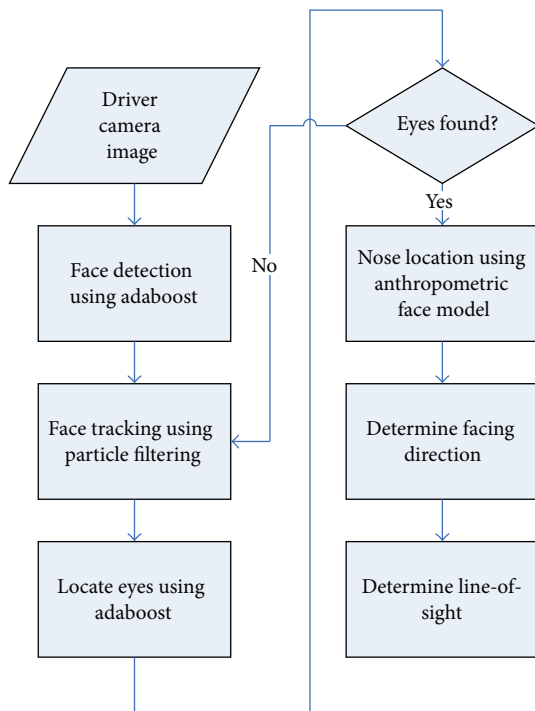


FIGURE 3: Flowchart for driver's line-of-sight analysis.

within an anthropometric face model [9, 10], and the location of the nose can be verified by the location of the nostrils. Because the camera is positioned directly in front of the face of the driver, the geometry form by the eyes and the nose is an equilateral triangle when the driver is facing forward, as shown in Figure 5. If the driver's head turns away from facing front, the triangle would no longer be equilateral, and the change can be used to calculate the facial orientation.

In Figure 5, the ratio of the lengths of the line segments "a" and "b" is used to estimate the turning angle of the head. For example, if the length of line segment "a" is greater than that of line segment "b," then it can be assumed that the driver's head is turning toward his right side. The relationship between these line segments and the facial orientation can be estimated using the relationship shown in Table 1. The pupil's location within its eye is used to adjust focus angle from  $-\pi/4$  to  $+\pi/4$  radians, as illustrated in Figure 6.

3.2. Lane Boundaries and Types. Images from the front facing-camera are used for detecting road conditions; which include the lane boundaries and whether they are crossable or not. However, in order to detect parallel lines and ascertain their types, an inverse perspective transform would need to be applied first to the images. Once the inverse perspective

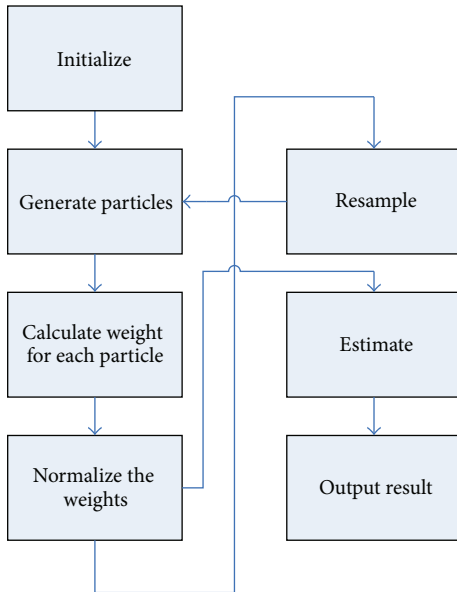


FIGURE 4: Flowchart for the particle filter.

TABLE 1: Conversion between the proportion  $a/b$  and corresponding angles.

$a/b$	$\approx 1$	$\approx 1.6$	$\approx 2.5$	$\approx 5$	$\approx 7$
angle	0	$\pi/16$	$\pi/8$	$3\pi/16$	$\pi/4$

transforms have applied, the color information is removed from the images, then a line-detector like Hough transform is applied to the images that should yield parallel line segments, and lines that are not long enough to be lane boundaries are then eliminated. Once the boundaries lines are detected, the center of the driving lane is calculated. An example is shown in Figure 7(c), where the blue line shown is the calculated lane center, which is in the middle of the boundaries lines. However, this method is known to fail in the case where the lane markings became faded or overlaid with artifacts created by, for example, tire tread. Because of this, false alerts can be issued in the current implementation. Future research will seek to deal with these problems

Based on the local traffic laws in Taiwan, lane boundaries can be one of three types: double solid (DS), single solid (SS), and single dashed (SD). The line segments detected would need to be assigned into one of these three classes using the distance separating the line segments. The lane boundary formed by double solid lines is a traffic divider, which is an uncrossable boundary; any deviation toward it will be penalized by the legal-maneuver analysis module our system, which will be discussed in a later section. The lane boundary formed by a single solid line is a lane divider, separating traffic moving in the same direction; deviations toward it are penalized, though not as heavily as the double solid line. The lane divider formed by a dashed line is also a lane divider, but there is no penalty associated with crossing it.

**3.3. Front Vehicle Detection.** The vehicle detection modules decide whether a vehicle exists and estimate the distance to

it, which is then normalized by the safety distance. Initially, we decided on the Haar-feature like Adaboosted classifier as described in [6]. We test several features as training input to the Adaboosted classifier for detecting vehicles and found that a good trade-off feature: the horizontal edge information. Figure 8 shows the process of extracting edge information from an input image, then keeping only the horizontal edges. As it can be seen, the basic outline of the vehicle is entirely visible throughout this process. So for an acceptable tradeoff between detection speed and accuracy, we chose to use the horizontal edge information. However, instead of the standard Adaboost algorithm, which assigns the same cost for misclassification of samples from different classes, the Cost-Sensitive (CS) Adaboost algorithm is selected instead for this application [11]. The CS Adaboost algorithm assigns different costs to samples from different classes, and the update weights of misclassified positive samples will be assigned greater cost than the misclassified negative samples. We trained our classifier with 267 positive samples and 1074 negative sample of vehicle images under various weather conditions, which were sufficient for the initial test. Once a vehicle is detected, the distance to it would need to be calculated or estimated.

Again, when deriving the distance-to-vehicle value, the tradeoff between speed and accuracy must again be considered. And considering that an accident may occur in a split second, we decided that a rough estimate based on the size of the detection box surrounding a detected vehicle is good enough. Figure 9 shows a real-world example of making such an estimate.

**3.4. Blind-Spot Vehicle Detection.** The blind-spot region next to the driving vehicle can be defined as triangular in shape (approximately a right triangle) with the length of the opposite side at 15 meters across and 6 meters across the adjacent side, as illustrated in Figure 10. The blind-spot detector cameras have narrow field-of-view, in order to focus only on objects located within the blind-spot regions. Figure 11 shows a sequence of images of a vehicle passing through the blind-spot region.

For the purpose of vehicle detection within the defined region during the day, three methods are implemented in parallel then a union operation is performed on their results. The first method is the sequential image subtraction using the hue-only colorband, this method works the best against a relatively uniform background, as when driving on a highway. The hue-only subband is selected because it is desirable that the subtraction operation is luminance-independent. However, a basic subtraction operation can result in a noisy image, so the resultant image is first denoised and thresholded before the largest connected component is extracted. Figure 12 shows the hue-based difference images of the sequence in Figure 11 with the largest connected component marked. In order to ascertain that the object is a car passing through the blind-spot area, the size of the largest connected component must exceed a preset threshold.

The second method utilized the standard Hough circle transform to detect the tires of the vehicle in the lower half of the images. In order to ascertain that the vehicle is within the blindspot region, the diameter of the detected circle(s) must



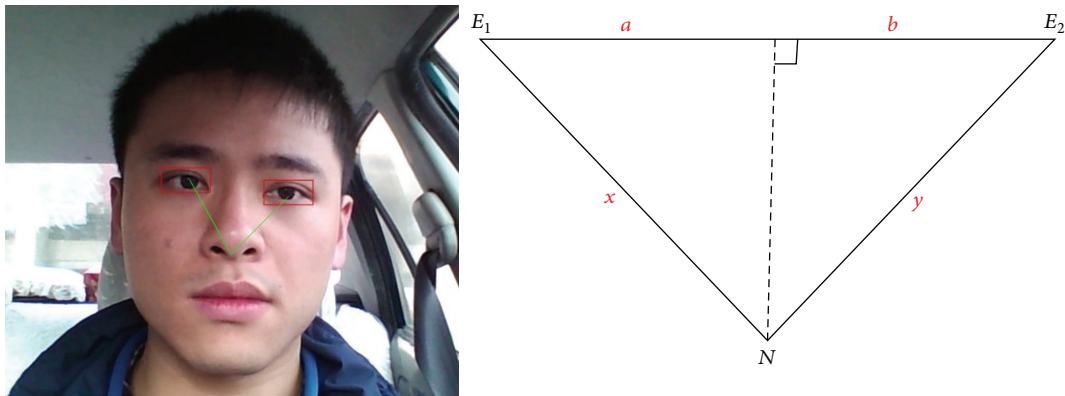


FIGURE 5: Diagram of the geometry formed by the driver's facial features.

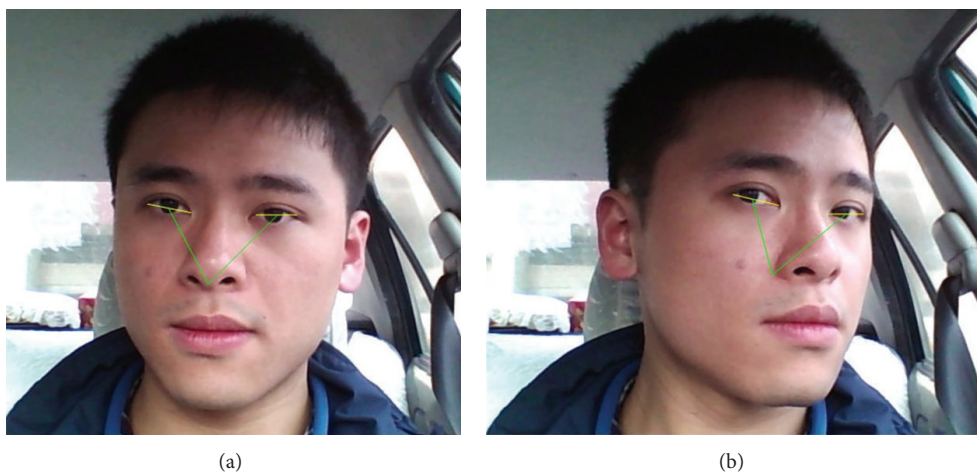


FIGURE 6: (a) Gaze estimation based on face turn angle. (b) Adjusted gaze estimation.

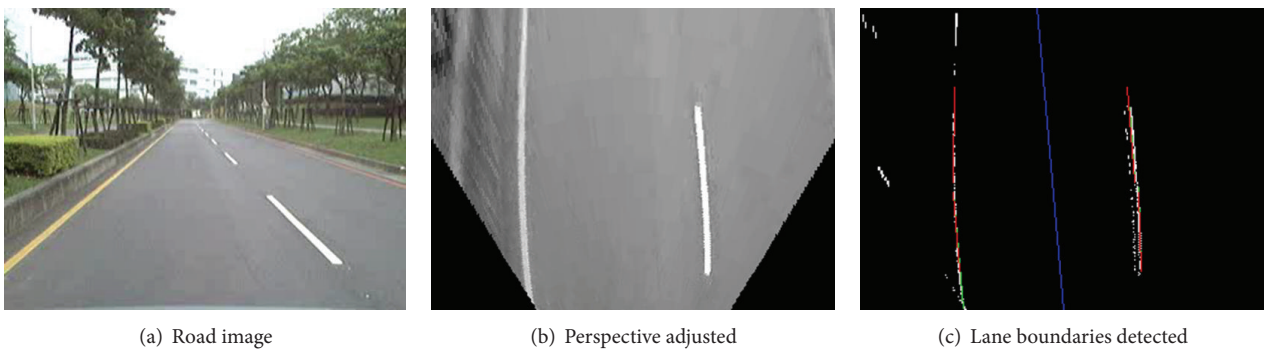


FIGURE 7: Detection of boundary lines.

meet a minimum size. Figure 13 shows successful detection of the front tire of the passing vehicle shown in Figure 11.

The third day-time method is to use the Haar-like Adaboost classifier trained using the edge information. Size and location of the detection box are used to determine whether the detected vehicle is within the blind-spot region. The results of these three methods are combined using a

union operator in order to ascertain that a vehicle is definitely within the blind-spot region.

For purpose of vehicle detection at night, the vehicle headlights are used as hint. First the contrast of each image is enhanced and then thresholded before connected regions in white in the lower half of the image are extracted. The size of the connected regions must meet a preset threshold to be

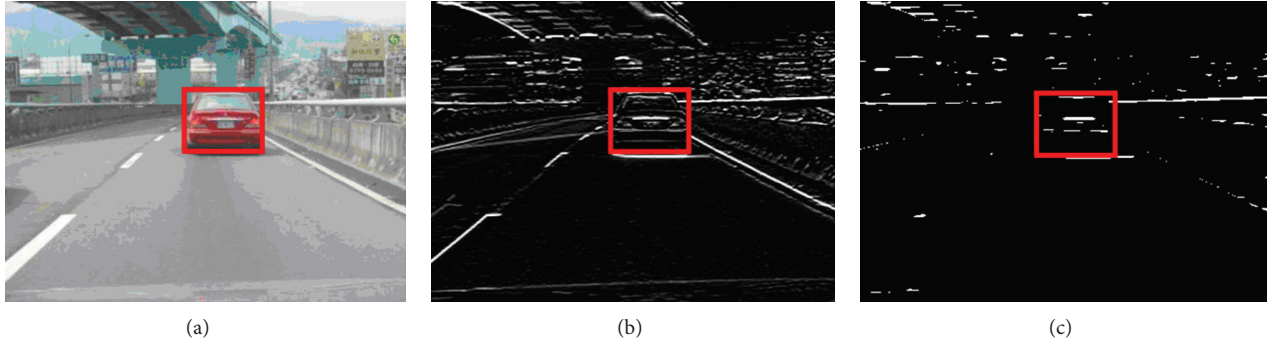


FIGURE 8: (a) Image of car, (b) horizontal edges, and (c) after morphology operation.

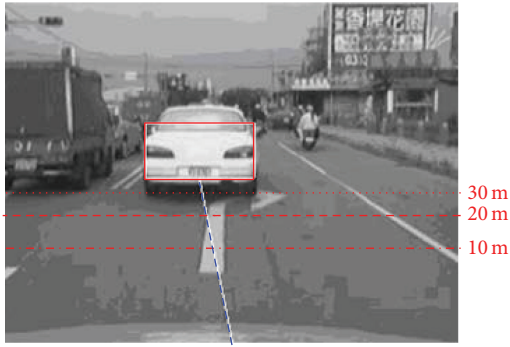


FIGURE 9: Estimation of distance-to-vehicle based on size of detection box.

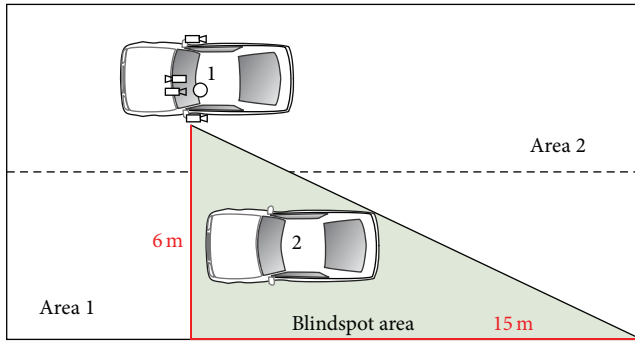


FIGURE 10: The definition of blind-spot regions.

determined as the headlights from a vehicle in the blind-spot region. Figure 14 shows a sequence of a passing vehicle at night, and Figure 15 shows the successful detection results.

#### 4. Data Analysis

The idea of automatic fuzzy rules extraction [12, 13] is to distribute the entire input space into overlapping intervals, then, according to some grouping methods, output the fuzzy rules for classifying these values and then finally defuzzify the results to obtain a crisp output value. Figure 16 shows a fuzzy inference system for extracting fuzzy rules. The minimum of each interval's maximum values is taken to form a basis for

generating the fuzzy rules, then they are defuzzified, which results in a crisp output value,  $y$ . The trapezoidal membership function is chosen as the membership function for our fuzzy inference system.

The equation in (1) describing the center-of-gravity method was chosen as the defuzzification method, where  $m$  is the mean within each interval,  $\sigma$  is the standard deviation, and  $d$  is the degree of membership. The output,  $y$ , is a crisp-valued indicator for the one of the three danger levels and is evenly distributed between 0 and 1:

$$y = \frac{\sum_{i=1}^n m_i \sigma_i d_i(x)}{\sum_{i=1}^n \sigma_i d_i(x)}, \quad (1)$$

where

$$d_k(x) = \begin{cases} 1, & \text{where } \mu < x \leq U, \\ 1 - \max(0, \min(1, r(\mu - x))), & \text{where } x \leq \mu, \\ 1 - \max(0, \min(1, r(x - U))), & \text{where } x > U. \end{cases} \quad (2)$$

The same fuzzy-inference structure is used for analysis under obstacle-less conditions as well as conditions in which obstacles exist. We will present each separately.

**4.1. Obstacle-Less Driving.** Three pieces of information are used as input features in this fuzzy-rule based safety analysis for obstacle-less driving module. These features include the amount of the vehicle's deviation from lane center,  $L_{Dev}$ , the speed of the vehicle, and the deviation of driver's line-of-sight from the lane center in angles. The definition of  $L_{Dev}$  can be seen in Figure 17 and (3):

$$L_{Dev} = \frac{(\text{Vehicle}_{Center} - L_{Center})}{\text{Width}_{Lane}} \quad (3)$$

The fuzzy-rules-based analysis system fuzzified each piece of these input information into three levels, "low," "medium," and "high," and output a warning based on the interactions of these variables. For example, if the driver's focus matches the vehicle's movement at a safe speed, as the scenario illustrated in Figure 18; then, the fuzzy inference



FIGURE 11: Sequence of a vehicle passing through the blind-spot region.

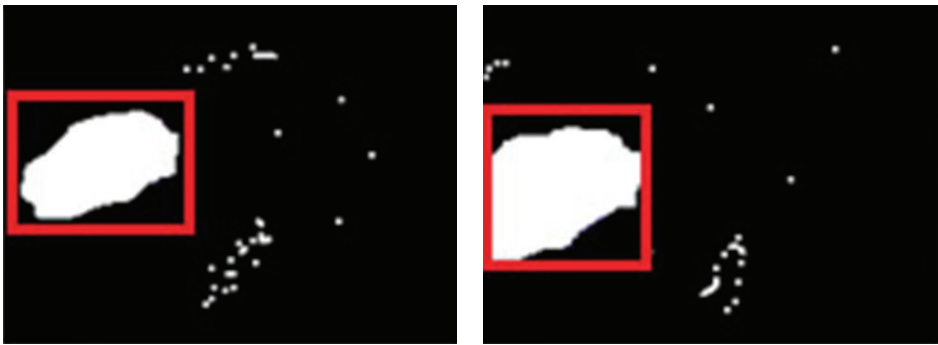


FIGURE 12: Results of hue-band image subtraction of the sequence in Figure 11.

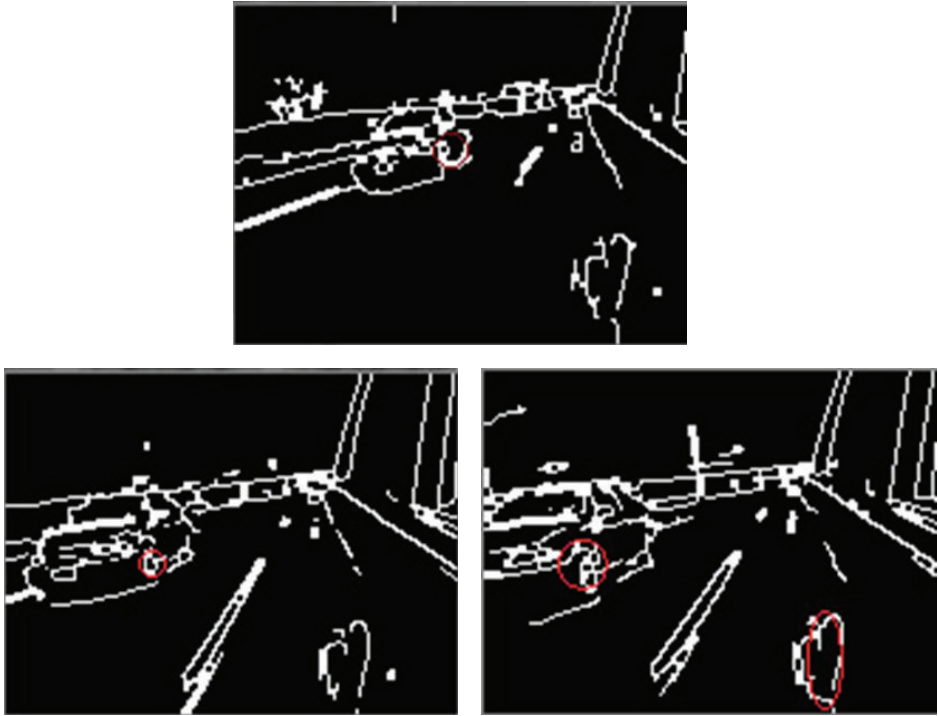


FIGURE 13: The successful detection of a front tire using Hough circle transform.



FIGURE 14: Sequence of a vehicle passing through the blind-spot region at night.



FIGURE 15: Successful detection of headlights of the vehicle in Figure 14.

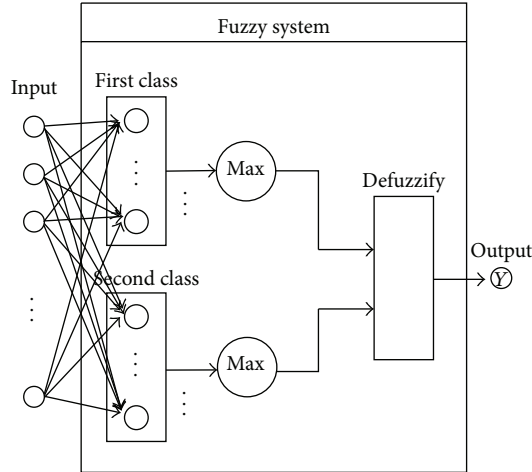


FIGURE 16: Diagram of the fuzzy inference system.

system would issue a “safe” level of danger, and no action would be taken to alert the driver. However, if the driver’s focus mismatches the vehicle’s movement for longer than a short duration, for example, for about 6 video frames, then even at a relatively safe speed the fuzzy inference system would issue a “risky” warning to the driver to refocus from his current distraction, as illustrated by the scenario in Figure 19.

**4.2. Driving with Obstacles Present.** The input information for the fuzzy-rules-based analysis for driving in the presence

of obstacles include  $L_{Dev}$  as defined by (3), the distance to the obstacle expressed as a percentage of the safe distance, and deviation of the driver’s line-of-sight from the obstacle expressed in angles. Each of these pieces of information is first fuzzified before being analyzed by the fuzzy-rules-based analysis system which then output a warning based on their interactions. For example, in Figure 20,  $L_{Dev}$  was low and the distance-to-vehicle was about 150% of the safety distance, but the driver’s focus is largely out of alignment with the obstacle, so ultimately the module issued a “risky” warning. Table 2 shows some of the interactions of these variables and their output.

**4.3. Blind-Spot Detection.** The blind-spot detection fuzzy-rules analysis module takes as input: include  $L_{Dev}$ , whether a vehicle exists in one of the blind-spot regions and the driver’s gaze angle. The inputs are also fuzzified before being used for analysis. Table 3 shows some of the input and output for this fuzzy-rules analysis module. The value for whether a vehicle exists in one of the blind-spots is either 0 or 1, for false or true, respectively. The output for when no vehicle is detected is always “safe.”

**4.4. Legal Maneuver Analysis.** The legal-maneuver analysis module is a crisp rule based analysis module. It takes as its input the lane boundary types and the  $L_{Dev}$  value. Its rules are simple: it checks the  $L_{Dev}$  value for the direction the vehicle is moving against whether the nearest boundary is crossable. If  $L_{Dev}$  value is low, the vehicle is not obviously



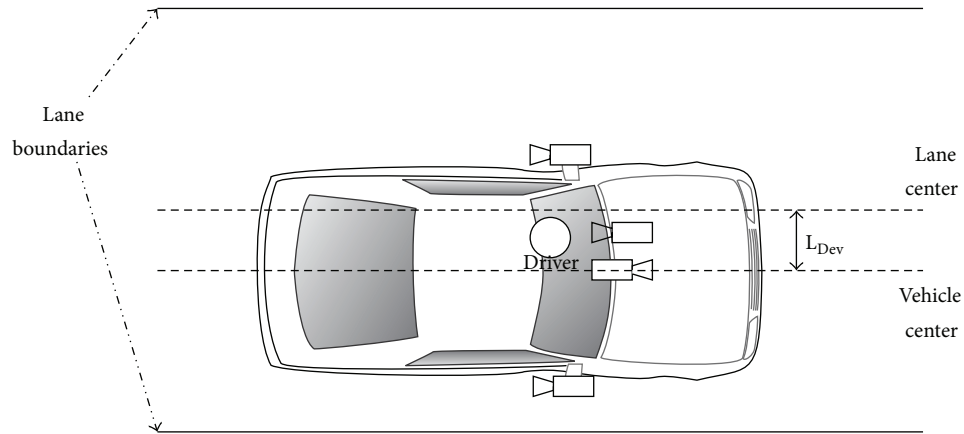
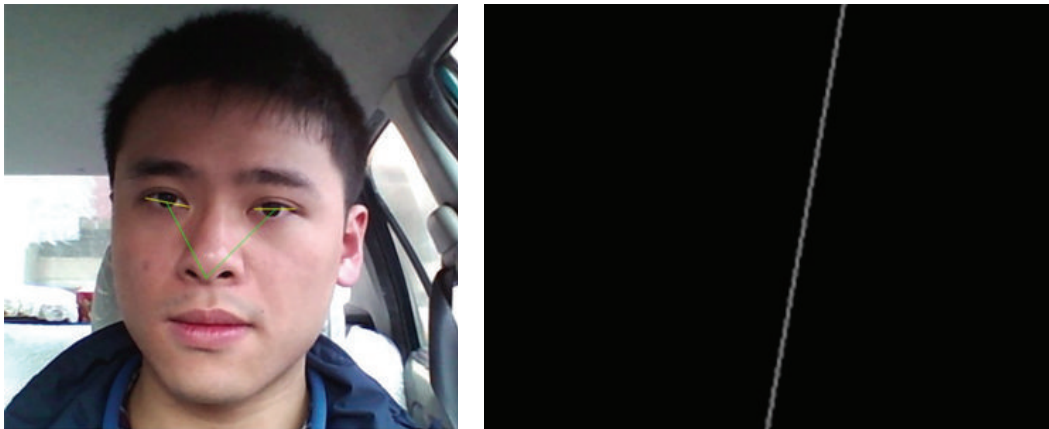


FIGURE 17: Graphical definition of lane deviation value,  $L_{Dev}$ .



(a) Driver's image and detection of driver's viewing angle



(b) Road image and detection of lane boundaries and center

FIGURE 18: Scenario where driver's focus matches vehicle movement.

moving toward either boundary; then, a "safe" is its output. If  $L_{Dev}$  value is relatively high, then either a "safe," "warning," or "danger" will be issued, depending on whether the boundary type is "single-dashed," "single-solid," or "double-solid," respectively.

### 5. Simulations

The parameters from several real-world driving situations were used as input to our system to test the system's performance. One of the simulations is presented in Figure 21. In

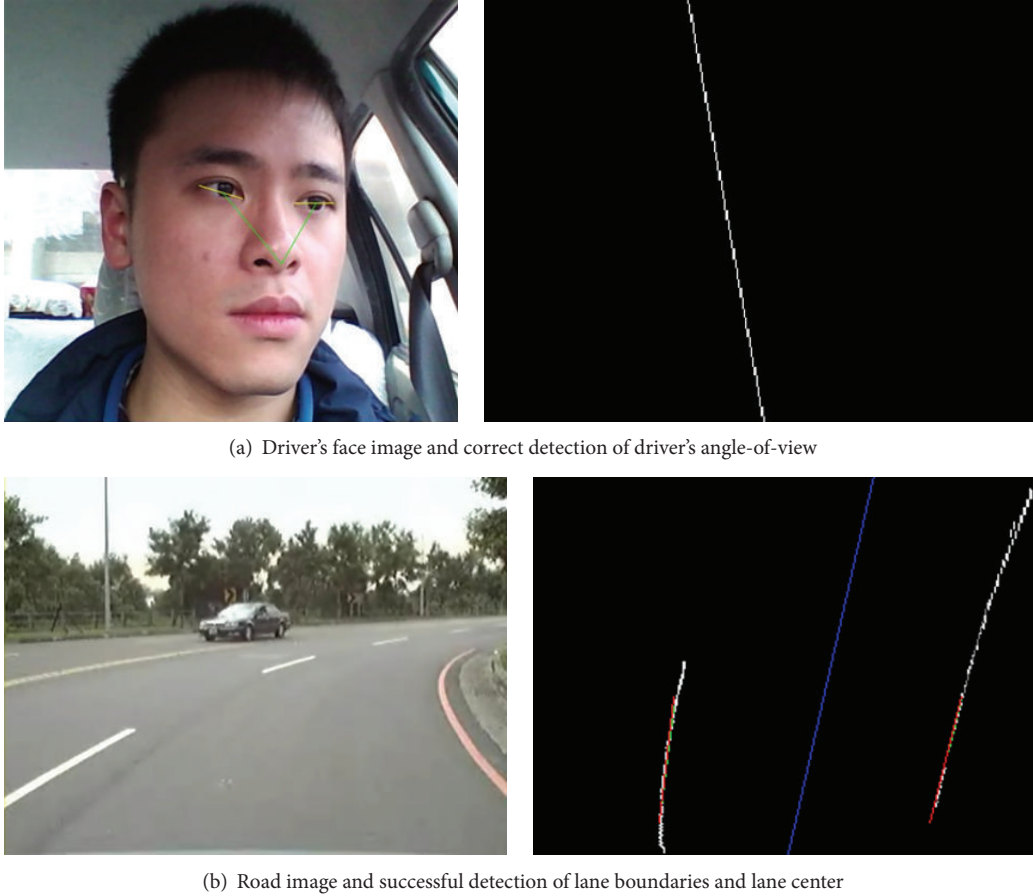


FIGURE 19: Scenario where driver's focus mismatches vehicle maneuver.

this simulation, the road ahead is empty of cars, the right side is a solid boundary, and the left lane is a crossable boundary with no vehicle in the neighboring lane, and the driver has intent to shift to the left lane.

Figure 22 shows the data flow data for the scenario in Figure 21.

Because no vehicle was in front of the car, the flow chart above shows that two of the routines were crossed out due to insufficient input data, so there was not sufficient number of inputs into the fuzzy-rules analysis in the presence of obstacles module, which automatically defaults its output to "safe" regardless of the other input values, as designed. The fuzzy-rules analysis in the absence of obstacles analyzed the driver's intent to move into the next lane with the car's deviation from the lane center and found the move to be "safe." The legal-maneuver analysis modules found the left lane boundary is crossable when the vehicle deviated toward the left, so it also output a "safe" signal. Finally, the fuzzy-rules analysis for blind-spot detection accepted that no vehicle existed in either blind-spot region, so it also output a "safe" signal, so the resultant system warning was "safe."

The results of other simulations were as expected and verified that system performs like the flowchart in Figure 2. In the next section, we will compare the performance of certain

TABLE 2: Some of the input and output of fuzzy-rules when vehicle is present.

$I_{Dev}$ (%) (- left, + right)	Normalized distance	Focus deviation	Output
4	2.13	5	Safe
-5	2.32	9	Safe
-7	3	3	Safe
8	2.01	10	Safe
13	0.97	17	Warning
32	0.13	52	Danger

parts of our system with others methods and present the results.

## 6. Experimental Results

In this section, experiments were performed to test each of the nonanalysis modules within our system worked as expected before comparisons were made to compatible algorithms. We have verified through simulation that our system performs as expected following the flow chart in Figure 2. Then, we can compare the performances of some of the more

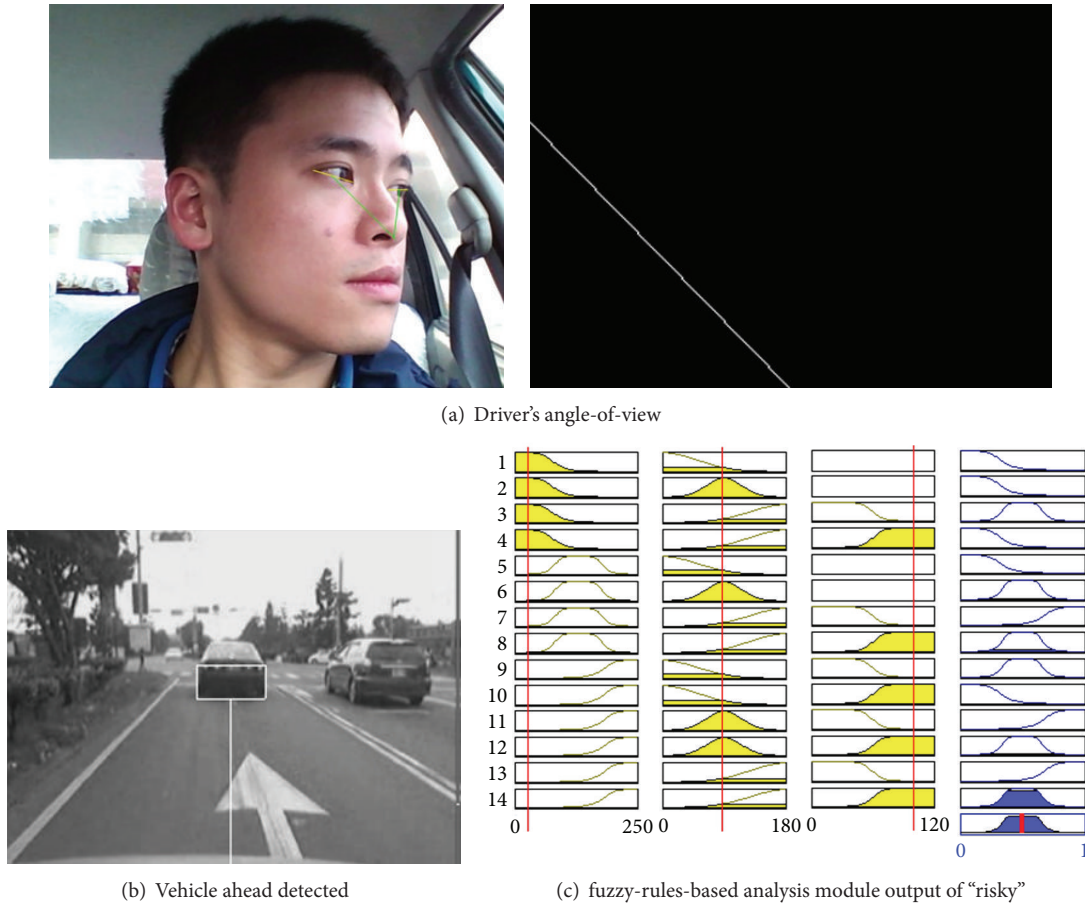


FIGURE 20: A scenario where driver's focus is misaligned with the vehicle ahead.

TABLE 3: Some of the input and output for the blind-spot fuzzy-rules analysis module.

Vehicle at right blind-spot	Vehicle at left blind-spot	$L_{Dev}$ (– left, + right) (%)	Gaze angle	Output
0	0	–5	–11	Safe
0	1	–19	–4	Warning
0	1	10	15	Safe
1	0	8	12	Warning
1	0	–20	–5	Safe
1	1	–32	–22	Danger



FIGURE 21: One of the driving scenarios: changing lane with no vehicle in blind-spots.

important modules within our system with other comparable methods.

The tests include the performance of the particle-filter based face tracker against the performance of a trained Haar-like feature Adaboosted classifier; the performance our trained cost-sensitive Adaboosted classifier will be compared to a standard Adaboosted classifier; and the performance for the fuzzy-rules-based analysis module for obstacle-less driving against a trained artificial neural network. Total of three comparisons were made.

Two video sequences were used for testing the performance of the face trackers. The first sequence, “A,” contains 372 frames of the driver’s face in various degrees of moving, tilting, and turning. The second sequence, “B,” contains 476

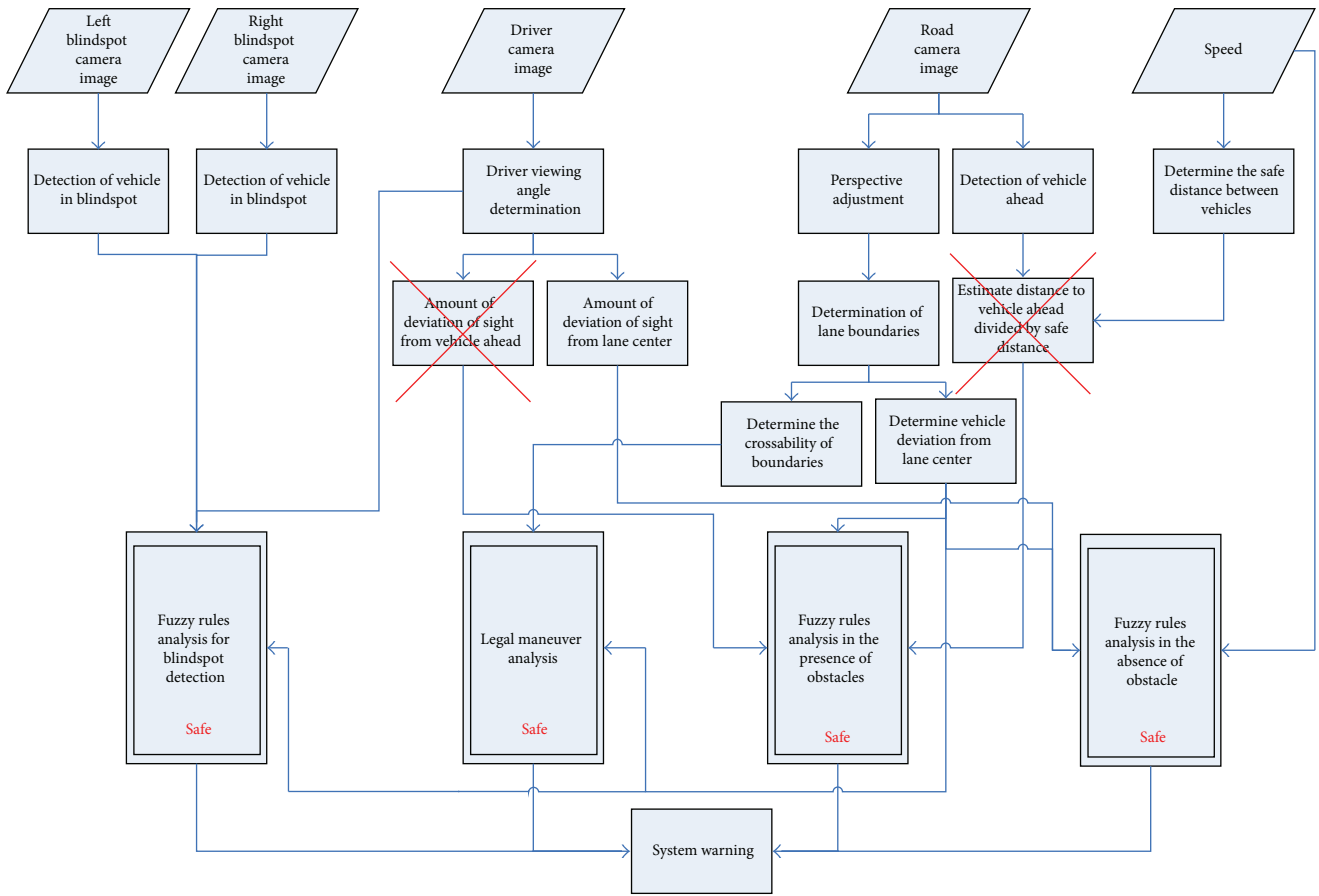


FIGURE 22: The flow chart for the driving scenario in Figure 21.

TABLE 4: Performance comparisons of head trackers in terms of number of correct frames (out of total number of frames).

Film sequence	Adaboost classifier	Adaboost classifier + particle filter
A	213 (372)	365 (372)
B	439 (476)	467 (476)

frames of the driver’s face yawning, partially occluded by small objects, mouth moving, and eyes closed temporarily. Table 4 shows the comparisons of the performances of both systems using these two video sequences. The number in quote is the total number of frames, and the unquoted numbers indicate how many frames each tracker has correctly detected the human face. It is obvious that the particle filter tracker outperformed the Adaboosted classifier in both sequences; with a maximum difference of 98.1% accuracy against 57.3% accuracy for sequence A. It is also obvious that the performance of the particle filter method is more consistent and thus robust across both video sequences; performing at better than 98%, while the performances of Adaboosted classifier alone varied from 57% to 92%.

In the second test, the performance of a trained CS Adaboost classifier of vehicles is compared to a regular Adaboost classifier trained using the same positive and

TABLE 5: Performance Comparisons of CS Adaboost versus regular Adaboost.

	Vehicle recognition
Adaboost	90.15%
CS Adaboost	94.16%

TABLE 6: Ann training rules for deviation versus speed.

Danger level	Deviation		
	Small	Medium	Large
Speed			
Fast	Medium	High	High
Normal	Low	Medium	High
Slow	Low	Low	Medium

negative training samples. Table 5 shows the recognition rate (existence of vehicle correctly detected). As shown, CS Adaboost is clearly a better performer than regular Adaboost by up to 4%.

The performance of the fuzzy rules-based safety analyzer is compared to a feed-forward artificial neural network with two hidden layers. The training rules for the artificial neural network are shown in Tables 6 and 7. The car speeds are



TABLE 7: Ann training rules for Angle between driver’s line-of-sight and car’s direction of travel versus Speed.

Danger level	Angle between driver line-of-sight and car’s direction of travel		
	Small	Medium	Large
Speed			
Fast	Medium	High	High
Normal	Low	Medium	High
Slow	Low	Low	Medium

TABLE 8: Some of the rules determined ANN versus the rules of the automatic extracted fuzzy rules, bold lines where the two differ.

(a)

ANN Rules			
Speed (km/h)	Angle between line-of-sight and driving direction (degrees)	Deviation from lane center. (%)	Danger level
<b>40.5</b>	<b>24.5</b>	<b>35</b>	<b>Medium</b>
77.7	47.1	25	High
50.5	44.9	54	Medium
25.9	32.8	30	Low
<b>72.3</b>	<b>4.13</b>	<b>7</b>	<b>Medium</b>

(b)

Fuzzy rules			
Speed (km/h)	Angle between line-of-sight and driving direction (degrees)	Deviation from lane center. (%)	Danger level
<b>40.5</b>	<b>24.5</b>	<b>35</b>	<b>Low</b>
77.7	47.1	25	High
50.5	44.9	54	Medium
25.9	32.8	30	Low
<b>72.3</b>	<b>4.13</b>	<b>7</b>	<b>Low</b>

TABLE 9: Comparison of results of safety analysis between ANN and fuzzy rules, out of 343 total cases.

Safe analysis method	Accurate	False alarm/fail	Accuracy
ANN (9 hidden nodes)	312	31	90.9%
ANN (12 hidden nodes)	314	29	91.4%
Automatic fuzzy rules extraction	321	22	94.5%

TABLE 10: Results of blind-spot vehicle detector used on a video sequence.

Total frames	Frames with vehicle	Correctly detected vehicle	False alarms/ failures	Sensitivity
184	158	151	24	92.31%
80	26	25	52	96.15%

divided into: fast, normal, and slow. The angle between the driver’s line-of-sight and the car’s traveling direction are in degrees, either to the left, or to the right. However, for Table 7, we simply summarized this information into small, medium, and large, since the direction of turning is not applicable for this table.

The experiments for ANN were set up using two different settings for the number of nodes in the hidden layer, and they are 9 and 12. The resultant output rules determined by the automatic fuzzy rules extractor and the trained ANN’s using rules above are slightly different. A side-by-side comparison of some of the rules is shown above in Table 8. The differences in the rules, in bold, shows that the same input values for both ANN and automatic fuzzy-rules extractor have different output. Generally, those that caused the ANN to issue medium alarm are determined by the fuzzy rules as low level of danger.

A total of 343 test samples with varying driving parameters were used to test both the neural network and the automatic fuzzy rules extraction module. Table 9 shows the results of both systems: the ANN classifier issued more false alarms or even completely fails to detect an emergency situation, at least, in 29 out of the 343 cases, compared to only 22 miscalls by the fuzzy rules module.

To test the blind-spot vehicle detector, two video sequences were used. The first sequence has 184 frames containing a vehicle in 158 of them with a relatively uniform background, the other has 80 frames containing a vehicle in 26 of frame with more complex background, are used to test the blind-spot vehicle detector. Table 10 shows the results of the detection, which show that the blind-spot vehicle detector does not as well as can be expected; it yielded too many false

alarms, for vehicle driving with a complex background, and the detector has room for improvement in the future.

## 7. Conclusions

We have presented a vision-based driver-assistance system that uses fuzzy-rules based system to correlate and analyze the interactions between driver's view, the vehicle's movements, and the vehicle's surrounding. In order to ascertain the system works as a whole, we first tested each data extraction module to work as expected, these include the driver's viewing angle detection, the location of lane boundaries and their cross-abilities, the estimation for lane center and the amount the traveling vehicle deviates from it, and the detection of vehicle ahead and in the blind-spots. The data extracted were mostly satisfactory under normal conditions and can be used in the analysis modules for outputting requisite warnings based on the inputs, in order to help the driver avoid accidents. After we verified each data extraction module, we performed lab simulations to test the performance of the entire system using several real-world driving conditions, and the results of the simulations confirmed that our system can perform as expected.

We have also shown, by experimental results, that key parts of our system including the face tracker, fuzzy rules-based safety level analysis for obstacle-less driving, and the car detection modules have each performed better than comparable Adaboosted classifier, an ANN with 12 hidden nodes, and a regular Adaboosted classifier trained on the same set of samples, respectively. In summary, this paper presented a fuzzy-rules-based driver-assistance system that analyzes the interactions between the driver, the vehicle, and its surrounding and issues timely alerts that would keep the driver from entering into dangerous traffic situations. In addition, the blind-spot related modules would alert the driver when the vehicle deviates toward a neighboring already occupied by a vehicle. Lastly, it is also by design that this system is composed of loosely integrated modules, so it will be able to accommodate inclusion of additional modules in the future without problems.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

The work was supported by the Ministry of Science and Technology, Taiwan, under Grant NSC100-2221-E-182-047-MY3.

## References

- [1] J. C. McCall, D. P. Wipf, M. M. Trivedi, and B. D. Rao, "Lane change intent analysis using robust operators and sparse Bayesian learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 431–440, 2007.
- [2] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.
- [3] L. Fletcher and A. Zelinsky, "Driver inattention detection based on eye gaze-road event correlation," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 774–801, 2009.
- [4] M. M. Trivedi, T. Gandhi, and J. McCall, "Looking-in and looking-out of a vehicle: computer-vision-based enhanced vehicle safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 108–120, 2007.
- [5] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: state of the art and perspectives," *Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 1–16, 2000.
- [6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. I-511–I-518, December 2001.
- [7] X. Xu and B. Li, "Head tracking using particle filter with intensity gradient and color histogram," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '05)*, pp. 888–891, Amsterdam, The Netherlands, July 2005.
- [8] F.-E. Ababsa, M. Mallem, and D. Roussel, "Comparison between particle filter approach and Kalman filter-based technique for head tracking in augmented reality systems," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 1021–1026, May 2004.
- [9] J. Batista, "A drowsiness and point of attention monitoring system for driver vigilance," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems Conference (ITSC '07)*, pp. 702–708, Seattle, Wash, USA, October 2007.
- [10] A. S. M. Sohail and P. Bhattacharya, "Detection of facial feature points using anthropometric face model," in *Proceedings of the IEEE International Conference on Signal-Image Technology and Internet-Based Systems*, pp. 656–665, December 2006.
- [11] K. M. Ting, "A comparative study of cost-sensitive boosting algorithms," in *Proceedings of the 17th International Conference on Machine Learning*, pp. 983–990, 2000.
- [12] B. N. Nelson, "Automatic vehicle detection in infrared imagery using a fuzzy inference-based classification system," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 1, pp. 53–61, 2001.
- [13] S. Abe and M.-S. Lan, "Fuzzy rules extraction directly from numerical data for function approximation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 1, pp. 119–129, 1995.



# Hindawi

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

