Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

On-Board Vision System for Lane Recognition and Front-Vehicle Detection to Enhance Driver's Awareness

Shih-Shinh Huang, Chung-Jen Chen, Pei-Yung Hsiao, and Li-Chen Fu

Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

The objectives of this research are to develop a driving assistance system that can locate the positions of the lane boundaries and detect the existence of the front-vehicle. By providing warning mechanism, the system can protect drivers from dangerousness. In lane recognition, Gaussian filter, Peak-Finding procedure, and line-segment grouping procedure are used to detect land markers successfully and effectively. On the other hand, vehicle detection is achieved by using three features, such as underneath, vertical edge, and symmetry property. The proposed system is shown to work well under various conditions on the roadway. The vehicle detection rate is higher than 97%. Besides, the computation cost is inexpensive and the system's response is almost real time. Thus, the results of the present research work can improve traffic safety for on-road driving.

1. Introduction

In recent years, the traffic problem gets more and more serious due to increase of vehicles. Most traffic accidents were caused by the negligence of the drivers. In order to reduce the number of traffic accidents and to improve the safety and efficiency of the traffic, the researches on Intelligent Transportation System (ITS) have been conducted worldwide for many years. Intelligent vehicle (IV) system is a component of the ITS system, which aims to assist drivers in perceiving any dangerous situations earlier to avoid the accidents through sensing and understanding of the environment around the vehicle.

In this paper, we develop a detecting and warning system which is able to pick up the information about two most familiar on-road objects: lane and vehicle. There have been quite many researches involved in these topics, but most of the underlying researches were suitable for only some specific weather conditions. Here, the system proposed intends to enhance the target capabilities for more general weather and road conditions. By this enhancement, the resulting system may interest both automobile manufactures and buyers much more since it becomes more practical towards real application.

Up to now, there have been numerous research results falling into the field of lane recognition. In [1], the authors generate the bird's-eye view image of the road plane first by using Inverse Perspective Mapping (IPM) to remove the perspective effect. Next, it extracts the lane markers based on the road constraints and the lane marker's width. Another research work with different philosophy can be seen in [2] where a curve road model was proposed. In that work, a deformable template method is used to optimize a likelihood function based on the proposed model. However, that optimization algorithm cannot guarantee a sufficient accuracy without consuming huge computational resources. Therefore, this work is not suitable for the real time application.

Pomerleau [3] proposed a Rapidly Adapting Lateral Position Handler (RALPH) system, which constitutes an adaptive high-speed matching procedure to determine the lane's curvature and its lateral offset. Though the RALPH approach reduces computation cost for rapid response, it suffered from low precision and influenced by the insufficient parameters.

There is an additional approach [4] which combines the Hough Transform and the Line-Snake model. It first divides an image into a few sub-regions along the vertical direction, and then performs the Hough transform on each sub-region to obtain an initial estimation of the lane boundaries. Afterwards, the Line-Snake model is exploited to improve the results of lane boundary detection.

There have been numerous approaches proposed so far to perform vehicle detection. The intensity-based symmetry method [5] is often used in the car-following situation, as the rear of most vehicles is typically symmetrical. However, this method is limited to the strict car-following or situations when the object being dealt with displays some degree of symmetry. Besides, the method of symmetry may cause false alarms due to the objects with symmetry property on roadway.

In ARGO project, Broggi et al. [6] produced a symmetry map by combining the gray-level and horizontal-edge symmetry information. Then, the position of the vehicle's bottom is found by fitting a template to the edge map.

Kruger et al. [7] used the optical flow method to detect the moving object, because optical flow contains information about the motion of a camera relative to its environment. This method does not require a priori knowledge about obstacle shape, but the ego-motion of the vehicle needs to be assumed from a separate module.

Betke et al. [8] detected the large brightness changes caused by the passing car. For a distant car, they use the projection of horizontal and vertical edges for detection and calculate the correlation coefficient between the input image and the images in the database for verification.

By using the statistical approaches, Wu et al. [9] used a PCA classifier to recognize the vehicles without any vehicles' feature as heuristics. Before feeding the image to the classifier, the image preprocessing is needed. However, the huge database construction may be a very difficult work under various conditions for any types of vehicles.

In section 2, we will describe some preliminary knowledge. Section 3 explains the lane recognition procedure used in our system. In section 4, the vehicle detection algorithm based on sign pattern and shape properties of the vehicle is presented. Experiment results are demonstrated in section 5. Finally, we conclude our works in section 6 with some discussions.

2. Preliminary

2.1 System Procedures

The procedures of this system's running are as follows:

- **Location of Lane Boundaries:**
  First, the system locates lane boundaries in the images and calculates the physical position of lane boundaries with respect to the host vehicle.
- **Identification of Region of Interest:**
  Given the detected lane boundaries, the system identifies the region of interest, which is a small position of the visual image containing the two lane boundaries.
- **Detection of Vehicle Existence:**
  After identifying the region of interest, the system detects the existence of vehicles within the region of interest. If there is a vehicle which appears in front of our vehicle, we then calculate the distance from the host vehicle to the front-vehicle (Location Problem).
- **Safety Problem Detection:**
  According to the results of locating lane boundary's positions and of detecting the existence of vehicles, we can determine whether the vehicle is in a safe situation or not.

2.2 Location Problem

In our proposed system, it will detect the lane boundaries and the front-vehicle. However, the relationship between the detected objects and the host vehicle, such as relative distance and approaching rate, is still unknown. Here, we exploit DLT (Direct Linear Transformation) [10] method to get the relationship between the object coordinate and the image coordinate. Besides, we assume that the camera lens has no optical distortion and the vehicle is moving on the flat ground plane.

3. Lane Recognition

The objective of lane recognition is to locate the lane boundaries with the painted lane markers under various conditions.

3.1 Lane Marker Properties

By close observation, we can find out several properties of the lane markers. The properties are shown in the following:

- **Brightness:**
  Lane markers are generally brighter than the road surface even if they are with various any kinds of colors.
- **Slenderness:**
  Lane markers are normally slender with higher intensity.
- **Proximity:**
  Lane markers forming a lane boundary normally end-to-end closed to their neighboring ones.

In the first place, we extract lane markers by using the brightness and slenderness properties, and then group the neighboring feature points into a lane marker. After that, we use a straight line segment to represent each lane marker.

3.2 Noise Removal — Gaussian Filter

In order to reduce the noise effect, the Gaussian Filter is applied as the first step to remove noise. After applying the Gaussian filter, we can remove the ripples caused by noises in the profile illustrated in 0. Therefore, when there is a lane marker, the gray-level values in the profile of the filtered image will increase and then decrease, monotonously. Based on the brightness and slenderness properties, we can ensure the aforementioned characteristic after applying the Gaussian smoothing operator. We use the Peak-Finding Algorithm to get the lane marker candidates in the profile of the filtered image. The Peak-Finding algorithm is described in the sequel.

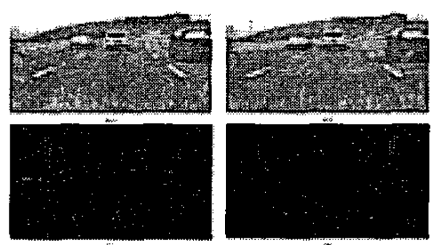


Figure 3.1: The effect of Gaussian smoothing operator

3.3 Feature Extraction — Peak-Finding Algorithm

The objective of this algorithm is to find the peaks in a scan line of the filtered profile.

0 is a part of 0(d). Referring 0, there are several "hills" separated by vertical lines. A formal definition of the "hill" can be given as:

A range over which the values increase first and decrease next without any internal ripples in the profile.

In other words, the profile value will first increase and then decrease monotonously when traversing a monotonous hill. Several parameters below are defined to formulate the monotonous hill:

- **Start Position p_s:** the position of first increase in the profile value
- **Start Value v_s:** the profile value at the start position
- **End Position p_e:** the position of last decrease in profile value. (Note that, pe is also the start position of another hill.)
- **End Value v_e:** the profile value at the end position
- **Peak Position p_p:** the position of first decrease in profile value
- **Peak Value v_p:** the profile value at the peak position. (The maximum value over a hill)

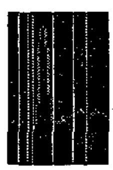


Figure 3.2: Several hills in the profile

0(a) shows the definition of the hill parameters in the schematic form. According to the predefined parameters of the hill, we can calculate the height and width of a hill illustrated in 0(b) and exploit such information to determine weather it is a real peak point corresponding to a lane marker point or not. Based on the brightness and slenderness properties, the parameters of *Left_Height* and *Right_Height* should be greater than a predefined threshold and the parameter of *Width* should be lower than a threshold.

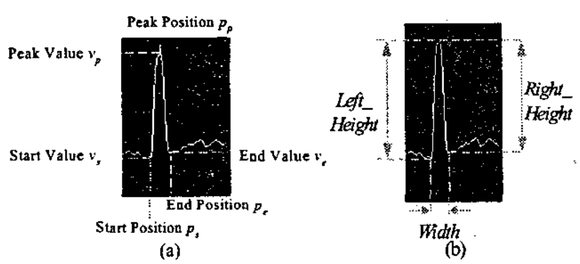


Figure 3.3: The parameters of a hill

After applying the proposed method, we can obtain the feature points as shown in 0. This image is named as Peak-Point image.

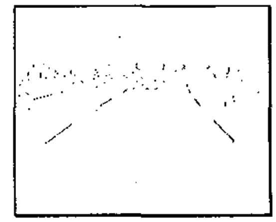


Figure 3.4: Peak-point image

3.4 Feature Point Grouping

After constructing the Peak-Point image, we obtain the lane marker candidates by aggregating the neighboring points. Incidentally, a lane marker candidate is named as a Line-Segment, which is defined as:

Line-Segment: $L(p_L(u_L,v_L), p_U(u_U,v_U), b_0, b_1)$,

where

p_L: Lower point of Line-Segment;

p_U: Upper point of Line-Segment;

b_0: The intercept of Line-Segment;

b_1: The slope of Line-Segment;

$p_x(u_x,v_x)$: u_x and v_x are horizontal and vertical coordinates, respectively.

However, noise points will make the direction of a Line-Segment diverge. In order to represent a Line-Segment in a more precise way, a least-square method is used to characterize the Line-Segment.

3.5 Line-Segment Combination

As we know, lane markers belonging to the same lane boundary will be subject to end-to-end adjacency. 0 shows two Line-Segments will be combined via this method. Solid lines represent Line-Segments and dashed lines are the extended lines generated from the lower Line-Segments' extended lines. The method is based on the fact that lane markers belonging to the same lane boundary should be lined up as much as possible. The lane recognition result is shown in 0.

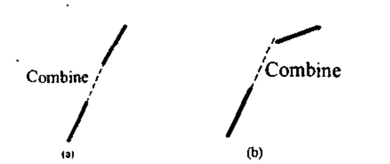
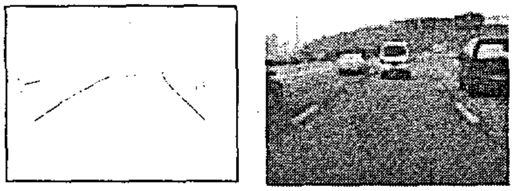


Figure 3.5: Combination of two Line-Segments

4. Vehicle Detection

The objective of vehicle detection is to detect any kinds of vehicles under various conditions. In our proposed method, three features, underneath, vertical edges, and symmetry property, are employed to detect and verify the vehicles due to the features' presence of a vehicle under various conditions. The feature properties and the method of feature extraction will be described in the subsequent sub-sections.



(a) (b)

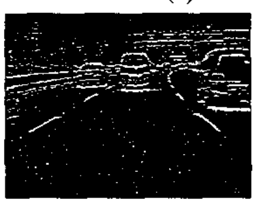
Figure3.6: Lane recognition result

4.1 Sign Pattern of Vehicle – Underneath

The concept of sign pattern was proposed by Mori et al. [11]. "Sign pattern" is defined as a pattern of environment and is neither necessarily a real object nor represented by attributes of static characters. In our proposed system, the sign pattern of vehicles is the dark shadow of underneath. Instead of thresholding the image, we extract the underneath candidates by checking the gradient of intensity as the local feature of underneath in the image. Because the underneath is spatially continuous, a mask like the sobel edge operator can be used to detect the underneath points. The mask is shown in 0(a) and the resulting image after applying underneath detection is shown in 0(b).

-1	-2	-1
1	2	1

(a)



(b)

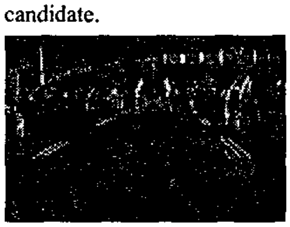
Figure4.1: The mask and the resulting image of underneath detection

4.2 Vertical Edges of Vehicles

The edge information is an obvious feature for vehicle detection. 0(b) shows the resulting image after applying vertical edge operation via the mask as shown in 0(a). Each vehicle image lies between two vertical edges. Based on this observation, the region between a pair of vertical edges will be taken as a vehicle candidate.

-1		1
-2		2
-1		1

(a)



(b)

Figure4.2: The mask and the resulting of vertical edge operation

In order to find the pairs of vertical edges, we calculate the histogram of vertical edge projection as illustrated in 0. A vertical edge will result in a peak in the histogram. As mentioned in lane marker extraction section, the Gaussian smooth operator and the Peak-Finding algorithm are applied to find out these peaks in histogram.

Furthermore, the height of the region for calculating vertical edge projection histogram is crucial. The height of region is the vehicle's height in the image. Hence, we must estimate the vehicle's height first in the image to determine the height of region. If vehicle's height is calculated more accurately, the peaks produced by the vertical edges in the histogram are more conspicuous.

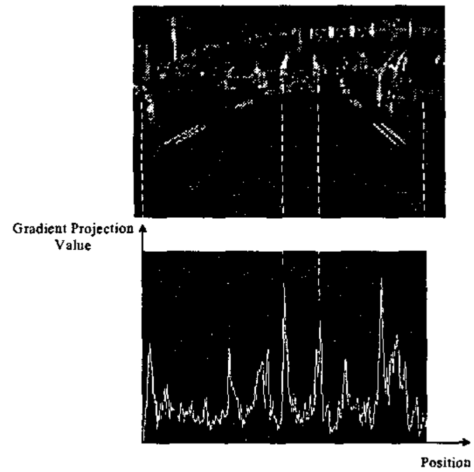


Figure4.3: The histogram of vertical edge projection in the region

4.3 Symmetry Property

Symmetry properties have been proved to be a strong feature for vehicle. In the proposed system, we define a window whose size is according to the typical aspect ratio of vehicles and perspective constraints. The gray-level symmetry score is defined as:

$$Gray\ level\ Symmetry = -\frac{\sum_{h=1}^{H}\sum_{w=1}^{W/2}\left|G(W/2-w,h)-G(W/2+w,h)\right|}{H\times W},$$

where $G(x,y)$ is the gray-level value of point (x,y), and H and W denote the height and width of the window, respectively. The higher symmetry value is, the more symmetric the region is.

4.4 The Procedure – Bottom-Up Searching Algorithm

After searching for the pairs of vertical edges, the underneath will be extracted for checking the vehicles' existence. For a pair of vertical edges, the width of the searching region in the image is the distance of the two vertical edges. The height of searching region is defined by estimating the bottom and top rows according to vehicle's width in the image. 0 shows the detected vertical edges and the searching region of underneath. Afterwards, the underneath searching is described below.

A flag array illustrated in 0 is used to determine the existence of underneath. The size of array is the width of vehicle candidate. First, the cells of the array are set to 0. From bottom to top in the searching region, the cell of the array in the correspondent column is set to 1 if the pixel is a defined underneath pixel. This region is considered as the underneath of the vehicle if the ratio of cells being set to 1 is greater than a predefined threshold.

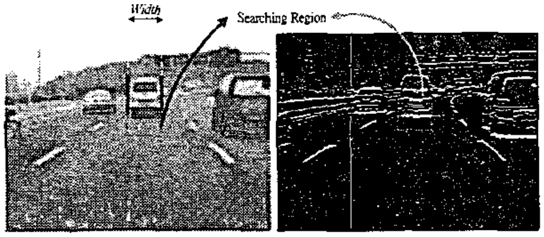


Figure4.4: The searching region

Figure4.5: The flag array for Bottom-Up Searching algorithm

5. Experiment

The hardware architecture of the system is shown in 0. The grabbed images are transferred to the ASUS mobile computer which is equipped with Intel Pentium IV 1.8GHz processor and 384MB RAM through IEEE1394 interface. We mount the digital camera on a real vehicle to continuously extract the traffic scenes as shown in Figure 5.2.

Figure 5.1: Image grabber and processing system

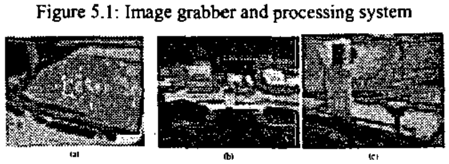


Figure 5.2 Experimental setup

5.1 Image Features under Various Conditions

The input images of the proposed system are obtained under various environmental conditions such as the normal daylight condition, the moist weather condition, and the condition inside a tunnel. In this study, two indexes are used to determine the condition of the environment. The first one is sharpness index, which can be measured by calculating the average of edge magnitude.

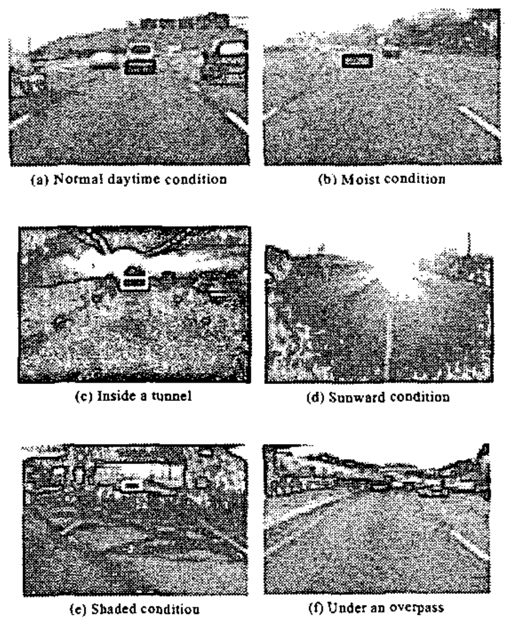
The second index is brightness index, which is used to measure the degree of darkness and determine whether it is daytime or night. This index is measured by calculating the average of intensity value in the image.

Table 5.1 The image indices under various conditions

Condition / Index	Daytime Normal	Moist	Inside a Tunnel
Sharpness Index	0.60	0.51	0.61
Brightness Index	0.64	0.64	0.45

5.2 Detection Result

The proposed lane recognition algorithm and vehicle detection is tested under various conditions illustrated in 0.



(a) Normal daytime condition
(b) Moist condition
(c) Inside a tunnel
(d) Sunward condition
(e) Shaded condition
(f) Under an overpass

Figure 5.3: Detection results

We test our vehicle detection algorithm in three conditions. In these images, the experimental results are listed in Table 5.2. In the normal condition, the system misses detection of the front-vehicle which is far away from our vehicle. In the moist weather condition, the edge information of the image is sometimes blurred and cannot be extracted. In the condition of inside a tunnel, the quality of image sometimes is bad because of the influenced lighting.

By reviewing the previous researches in the literature, we found that the vehicle detection rates are from 92% to 99%, which are very close to ours. However, the environments in their systems are either very specific or are subject to some constraints. Therefore, our proposed system, the detection rate is not only appealing for a driving assistance system but also the system's function can stand more practical circumstances.

Table 5.2 Vehicle detection rate under different conditions

	Daytime Normal	Moist Condition	Inside a Tunnel	Total
Hit	1174	655	131	1960
Miss	12	30	10	52
False Alarm	10	25	12	47
Detection Rate	98.9%	95.6%	92.9%	97.4%

5.3 Processing Time of System

The processing time in our proposed system is depicted in Table 5.3. The system is effective and only takes 70 ms to detect the lane and the front-vehicle.

Table 5.3 The Processing time of system

Initialization		10ms
Lane Recognition	Image Preprocessing	5ms
	Lane Marker Detection	10ms
Vehicle Detection	Image Preprocessing	10ms
	Vehicle Searching	35ms
Total		70ms

6. Conclusion

In this paper, we developed a lane recognition and vehicle detection system by adopting the computer vision technology. Our proposed system is robust even under various conditions and the computation cost of our system is quite economical.

For lane recognition, the Peak-Finding algorithm is proposed to extract the feature points effectively based on the lane markers' characteristics. Then, lane boundaries are detected successfully by grouping the feature points. On the other hand, vehicle detection is achieved by using three features of vehicles. First, the vertical edges of vehicles are exploited to search for the potential region of vehicles by applying the same method, Peak-Finding algorithm. Then, the Bottom-Up Searching algorithm is applied to detect the presence of underneath. Finally, the symmetry property is used to verify the vehicle candidates. The integration of the vehicle detection and lane recognition will reduce false vehicle detections to a great extent.

However, in this paper we only devoted our attention to detect the vehicle in front of the host vehicle. In the future, we will extend the work of lane recognition to the adjacent lanes.

Reference

[1] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, No 1, pp. 62-81, Jan. 1998.

[2] K. Kluge and S. Lakshmanan, "A Deformable-Template Approach to Lane Detection," *Proceedings of the Intelligent Vehicles Symposium*, pp. 54-59, Sep. 1995.

[3] D. Pomerleau, "RALPH: Rapidly Adapting Lateral Position Handler," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 506-511, 1995.

[4] D. J. Kang, J. W. Choi and I. S. Kweon, "Finding and Tracking Road Lanes Using "line-snakes"," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 189-194, 1996.

[5] Zielke T., Brauckmann M., and Von Seelen W., "Intensity and Edge-Based Symmetry Detection with an Application to Car-Following," *Image Understanding*, vol. 58, No. 2, pp. 177-190, 1993.

[6] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo Vision-based Vehicle Detection," *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, pp. 39-44, Oct. 2000.

[7] Kruger W., Enkelmann W., and Rossel S., "Real-Time Estimation And Tracking Of Optical Flow Vectors For Obstacle Detection," *Proceedings of Intelligent Vehicles Symposium*, 304-309, 1995.

[8] M. Betke, E. Haritaoglu, and L. S. Davis, "Multiple vehicle detection and tracking in hard real-time," *Proceedings of the IEEE Intelligent Vehicles Symposium*, Sep. 1996.

[9] Junwen Wu and Xuegong Zhang, "A PCA classifier and its application in vehicle detection," *International Joint Conference on Neural Networks*, vol. 1, 2001.

[10] Y. I. Abdel-Aziz, and H. M. Karara, "Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry", *Proceedings of the Symposium on Close-Range Photogrametry*, pp. 1-18, Falls Church, VA: American Society of Photogrammetry, 1971.

[11] H. Mori, N. Moghadam, and T. Matsushita, "On-Line Vehicle and Pedestrian Detections Based on Sign Pattern," *IEEE Transactions on industrial electronics*, vol. 41, No. 4, Aug. 1994.