Computers and Mathematics with Applications 61 (2011) 2096–2100

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Vision-based vehicle detection for a driver assistance system

Ying-Che Kuo [a,*], Neng-Sheng Pai [a,1], Yen-Feng Li [b]

[a] Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung, Taiwan

[b] Institute of Electronic Engineering, National Chin-Yi University of Technology, Taichung, Taiwan

ARTICLE INFO

Keywords:
Advance driver assistance systems (ADAS)
Vehicle detection
Optical flow

ABSTRACT

On the basis of a cost-effective embedded system, this work implements a preceding vehicle detection system by using computer vision technologies. The road scenes are acquired with a monocular camera. The features of the vehicle in front are extracted and recognized by the proposed refined image processing algorithm, and a tracking process based on optical flow is also applied for reducing the complexity of computing. The system also provides the longitudinal distance information for the further function of adaptive cruise control. Moreover, voice alerts and image recording will be activated if the distance is less than the safe range. A statistical base of 100 video road images are tested in our experiments; the natures of the vehicles include sedan, minivan, truck, and bus. The experimental results show that the proportion of correct identifications of proceeding vehicles is above 95.8%, testing on highways in the daytime. Experimental results also indicate that the system correctly identifies vehicles in real time.

1. Introduction

Advanced driver assistance systems (ADAS) have received considerable attention in recent decades, because many car accidents are caused mainly by drivers' lack of awareness or fatigue. Warning the driver of any dangers that may lie ahead on the road is important for improving traffic safety and accident prevention. So a major function of ADAS is the detection of vehicles in front of one's own by using computer vision technologies, since the cost of an optical sensor (such as CMOS and CCD ones) is much lower than that of an active sensor (such as laser or radar ones). Furthermore, optical sensors can be used in a wider range of applications, such as in lane departure warning systems and event video recorders.

Various vehicle detection approaches have been reported in the computer vision literature. Huang et al. [1] proposed a Gaussian filter, a peak-finding procedure, and a line-segment grouping procedure for detecting lane marks successfully. Then the vehicle detection is achieved by using features of undersides, vertical edges, and symmetry properties. In [2], Sun et al. proposed vehicle feature extraction and a classification method using a Gabor filter and a support vector machine (SVM). They used genetic algorithms (GA) to optimize filter banks, and used clustering to find the filters that feature parameters that are similar, and deleted redundant filtering. Khammari et al. [3] proposed gradient driven hypothesis generation and appearance-based hypothesis verification in order to implement the vehicle detection system. Some other literature entries [4–6] proposed learning algorithms for selecting the collection of features that best distinguish between vehicles and non-vehicles. However, the iterative training of classifiers and data set collection are complicated even if they can be done in advance. Besides, the detection procedures are complex and time-consuming, so most of the research implemented vehicle detection algorithms in a personal computer to achieve real time processing.

[*] Corresponding author. Tel.: +886 4 23924505x7271.
E-mail addresses: kuoyc@ncut.edu.tw (Y.-C. Kuo), pai@ncut.edu.tw (N.-S. Pai).

[1] Tel.: +886 4 23924505x7261.

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.08.081

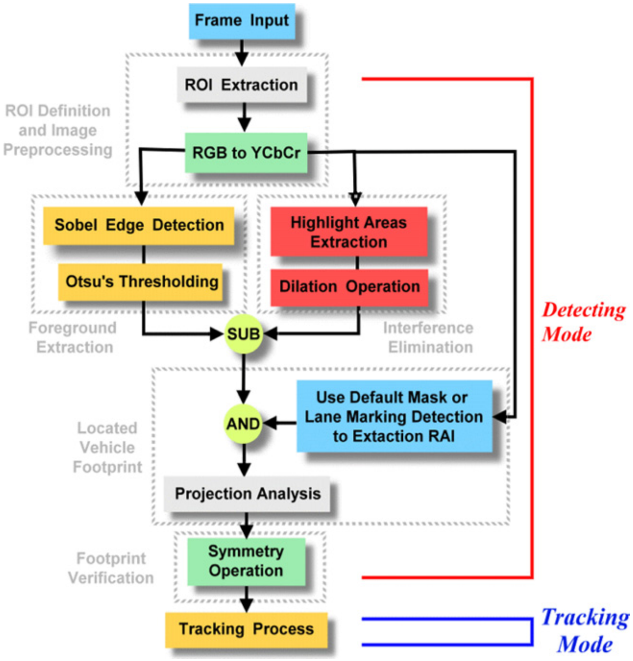


Fig. 1. Processing the flow of vehicle detection.

On the other hand, the motion-based methods [7–9] detect vehicles using optical flow to find the vehicles and analyze vectors to find and classify the moving objects in an image. However, this is also time-consuming and thus impractical for a real time system.

While implementing the vehicle detection system, the space in the car and the cost of hardware platforms are decisive considerations. Our work aims at proposing a refined vehicle detection algorithm that can be implemented in a cost-effective embedded platform. The vehicle detection algorithm includes road area finding, features of vehicle extraction, and vehicle verification. A tracking process dedicated to the detected vehicle region of the image based on optical flow is also applied to reduce the complexity of the computing. Furthermore, for the estimation of distance, a preceding vehicle range-finding method based on a geometric perspective has also been presented. Voice alert and image recording will be activated if the distance is less than the safe range.

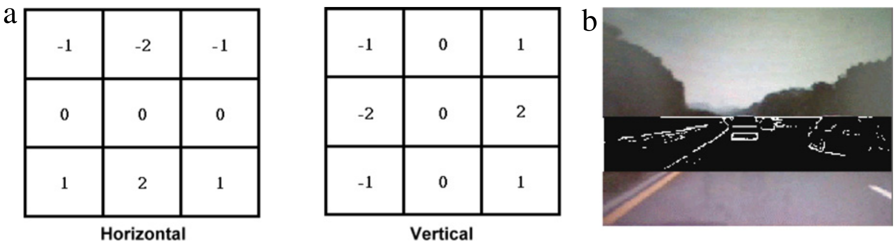
2. The vehicle detection algorithm

In our work, a CMOS camera is mounted in the interior on the windshield of a test car and the optical axis is parallel to the road surface. The camera captures the RGB color road environment image forward. For the considerations of realization in an embedded system with limited hardware resources, the procedures of our proposed refined vehicle detection algorithm are shown in Fig. 1 and described as follows. Firstly, the region of interest (ROI) area, the region of vehicles possibly appearing in the image, is defined. The left and right boundaries of the ROI are the left and right boundaries of the captured image. The top of the ROI is the horizon line in the image. The bottom of the ROI is 5 m in front of the test car in the image, in our experience. Consequently, many unnecessary influences (e.g., lamps, traffic lights, signs, etc.) outside the ROI are excluded. Next, to reduce the computing complexity, the color image of the ROI will be transformed from RGB color space to YC_bC_r color space. Only the luminance component Y of the image is required for subsequent image processes.

2.1. Profiles of vehicle extraction

Even though the road accounts for 50%–70% of the ROI, the ROI still includes large useless areas. The ordinary road surface includes lane marks, text on the road, seams of bridges and shadows of overpasses, all of which produce interference. The following methods are proposed for eliminating the interference and strengthening the profiles of vehicles.

Since the rear view of a vehicle is rectangular shape, Sobel edge filters with vertical and horizontal gradients [10], described in Fig. 2(a), are employed first to enhance the profiles of the rectangular objects in the ROI. Next, Otsu's threshold selection method [11] is applied to extract the foreground objects from the Sobel filter enhanced image which then yields the binary edge-detected image. At the same time, the road surface is sampled at five fixed areas, and the mean gray value of the five samples is calculated. The binary road image is obtained by binarizing the original ROI according to the mean gray value as a threshold. Then, the dilation operation [10] is applied subsequently to intensify the binary road image. Finally, the lane



-1	-2	-1
0	0	0
1	2	1

Horizontal

-1	0	1
-2	0	2
-1	0	1

Vertical

Fig. 2. (a) Sobel edge filter. (b) "No lane marks" image.

marks and interference can be removed using the binary edge-detected image and the binary road image in a subtraction operation, yielding an image with no lane marks (NLM), as presented in Fig. 2(b).

2.2. The range of the lane

An area in an image bounded by lane marks is called a road area image (RAI), where vehicles possibly appear. To verify the RAI area, the lane marks need to be detected in advance. In most images, lane marks are at 45° when they appear. Therefore, Sobel edge filters with ±45° gradients in the vertical are applied. Once the lane marks are enhanced by edge filters, the procedures of lane mark detection are described as follows.

(1) The start point of this detection is at the center in the horizontal direction and at Y_b (the bottom of the ROI) in the vertical direction.
(2) From the start point toward both sides in the horizontal direction, find the first bright pixels on both sides.
(3) Once the first bright pixels are found on both the left and the right sides, record the positions of the pixels found.
(4) If a bright pixel cannot be found when the horizontal position has already surmounted the last position that was found in the last finding, use one pixel inward of the last finding in the horizontal direction as this record.
(5) Move the start point to the center in the horizontal direction and effect a position increase of one pixel from Y_b toward Y_t (the top of the ROI) in the vertical direction. Repeat steps 2–5 until Y_t is reached in the vertical direction.

Once the procedure has been completed, the range of the lane is bounded by the record pixels and the RAI is found between the two lane marks.

2.3. The footprint of vehicle extraction

In this paper, we focus on the detection of preceding vehicles. The preceding vehicles will normally appear in the RAI of the captured image. In this step, the NLM image and the RAI image are used to execute the logical-AND operation; the result is called the footprint image. The footprint of the vehicle in the lane is obtained from the footprint image.

Locating the vehicle is the most important part of the detection process. Projection analysis is utilized herein to locate the left, right and bottom boundaries of the vehicle. The projections of the profile of the vehicle in horizontal and vertical directions are determined from the footprint image. The two highest peaks of the projection in the horizontal direction generally indicate the right and left boundaries of the vehicle, and are denoted as V_R and V_L respectively. The highest peak of the projection in the vertical direction usually indicates the bottom boundary of the vehicle, and is denoted as V_B. Then, the *vehicle image block* is found, with the boundaries. The width of the vehicle image block is $W = |V_R - V_L|$. The height of the vehicle image block is set as $H = 0.8W$.

2.4. Footprint verification

In Section 2.3, the footprint image may include wet spots, the seam of a bridge, and shadows of passing vehicles. Therefore, a rule that determines whether these footprints are associated with vehicles is required. In this step, a symmetry operation is applied to verify the footprints. The symmetry equation is modified [9] using Eq. (1). Our approach is to find the most symmetric axis by minimizing the symmetry measure $S(j)$ with the symmetry axis at $x = j$ within the vehicle image block:

$$S(j) = \sum_{i=V_B}^{V_B+H} \sum_{\Delta x=1}^{W/2} \sum_{j=V_L-\Delta k}^{V_R+\Delta k} |p(j+\Delta x, i) - p(j-\Delta x, i)| \tag{1}$$

$$j_{\mathrm{sym}} = \arg\min_j S(j) \quad \text{and} \quad \min S(j) < S_{\mathrm{th}}.$$

$p(j, i)$ denotes a component in the vehicle image block. V_B, V_L and V_R are the bottom, left and right positions of the boundaries of the vehicle image block, respectively. W and H represent the width and height of the vehicle image block. Finally, the vehicle image block can be identified as a vehicle if the minimum symmetry measure is less than a certain threshold S_{th}. The value of S_{th} depends on the location of the vehicle at different longitudinal distances in the image.

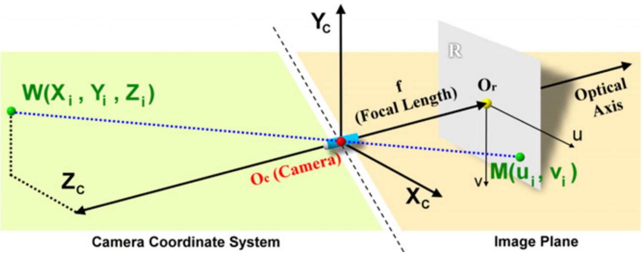


Fig. 3. Relation of the camera coordinate and the image plane.

3. The vehicle tracking process and distance estimation

3.1. Vehicle tracking

In the previous sections, the vehicle was located in the image plane. However, were every image frame to have to undergo such heavy and complicated processing, the burden on the embedded system would be prohibitive. Once the candidate vehicle is verified, optical flow is therefore applied herein to track the vehicle image block instead of executing the detection algorithm. The *sum of squared differences* (SSD) [12] with low computing complexity is utilized to implement the tracking process. The equation for the SSD is as follows:

$$\mathrm{SSD}(\delta x, \delta y) = \sum_{i=-\frac{W-1}{2}}^{\frac{W-1}{2}} \sum_{j=-\frac{H-1}{2}}^{\frac{H-1}{2}} [I_k(x+i, y+j) - I_{k+1}(x+i+\delta x, y+j+\delta y)]^2. \tag{2}$$

(x, y) is the center of the vehicle image block, where $-W/2 < \delta x < W/2$ and $-H/2 < \delta y < H/2$. I_k represents frame k and I_{k+1} represents frame $k+1$. The target is the vehicle image block, so the SSD involves performing the full search method only over a small block using a few operations. If the SSD is greater than the threshold, we re-execute the vehicle detection process in the next frame instead of the vehicle tracking process. Otherwise, we perform the tracking process continuously in order to reduce the computing complexity.

3.2. Estimation of the distance to the preceding vehicle

The position of the footprint of the detected proceeding vehicle on the road and the parameters of the camera are utilized to estimate the longitudinal distance to the preceding vehicle. We apply the estimation model presented in [13,14] in order to estimate the distance between the preceding vehicle and the test car from the captured image. A 3D scene point (X_i, Y_i, Z_i) in the camera coordinate system captured by the camera is projected onto the pixel (u_i, v_i) on the 2D image coordinate as shown in Fig. 3. Eq. (3) shows the projective phenomenon relation. f is the focal length of the camera. S_v is the scaling factor for the height (the ratio of the physical height and the image pixel). S_u is the scaling factor for the width (the ratio of the physical width and the image pixel). We set the camera's optical axis parallel to the road surface at a height H above the ground and assume a planar road surface. The underside of the vehicle on the road at a distance Z_i in front of the camera will project to the image at a height v_i. Then the distance Z_i can be estimated by using Eq. (4):

$$u_i = \frac{fS_uX_c}{Z_c}, \qquad v_i = \frac{f\,S_vY_c}{Z_c} \tag{3}$$

$$Z_i = \frac{fS_vH}{v_i}. \tag{4}$$

4. Experimental results

The proposed system was implemented on the INTEL XScale PXA270 SoC-based (520 MHz system clock, 32 MB Flash ROM, and 64 MB SDRAM) embedded hardware platform with a few peripheral devices (e.g., a TFT-LCD touch screen, Ethernet, AC-97, USB, etc.). A CMOS camera was connected to the hardware platform with a USB interface and mounted behind the windshield of the test car to acquire a QVGA (320 by 240) resolution image at 30 frames per second. The software system integrates the application program realizing the proposed algorithm in C language with V4L2 (USB camera driver), Madplay (AC-97 audio driver) and MiniGUI (TFT-LCD touch screen user interface driver) open source codes under the Linux operating system. The experimental environment was set on Highway No. 1 in Taiwan and the experiments were performed in the daytime with sufficient natural light. Fig. 4 presents the detection results for various kinds of interference. A statistical base of 100 vehicle video images are tested in our experiments; the natures of the vehicles include sedan, minivan, truck, and bus.

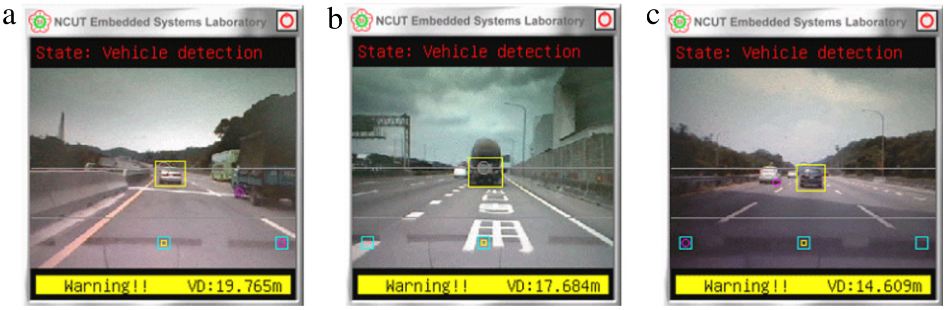


Fig. 4. Detection results with interference. (a) With a seam of a bridge. (b) With text on the road surface. (c) With a shadow under a viaduct.

The experimental results show that the proportion of correct identifications is above 95.8% in the detection of preceding vehicles. The proportion of correct identifications is above 90.6% in the detection of passing vehicles. The irregular rear view of a truck usually causes the false negative result in the detection. The processing time for vehicle detection is 0.16 s and that for vehicle tracking is 0.058 s on average. On the basis of the experimental results, we say that the proposed system can correctly identify most vehicles in real time.

5. Conclusion

This work presented a refined vehicle detection algorithm and tracking process that can be implemented in a cost-effective embedded system with limited hardware resources. The proposed system is successfully implemented on a test car and tested on Highway No. 1 in Taiwan, and its effectiveness is verified. The system also provides distance information for the further function of adaptive cruise control. Moreover, voice alerts and image recording will be activated if the distance is less than the safe range. We hope that the proposed system will be useful for building ADAS. For further studies, more environmental factors (e.g. the presence of strong shadows, harsh weather conditions, the lighting conditions depending on the time of the day, artificial illumination, etc.) must be considered to optimize system performance and make the system more robust.

Acknowledgement

The present work was supported by the National Science Council of Taiwan under Grant NSC-98-2221-E-167-018.

References

[1] S.S. Huang, C.J. Chen, P.Y. Hsiao, L.C. Fu, On-board vision system for lane recognition and front-vehicle detection to enhance driver's awareness, in: IEEE Int'l Conf. on Robotics and Automation, vol. 3, 2004, pp. 2456–2461.
[2] Z. Sun, G. Bebis, R. Miller, On-road vehicle detection using Gabor filters and support vector machines, in: IEEE Int'l Conf. on Digital Signal Processing, vol. 2, 2002, pp. 1019–1022.
[3] A. Khammari, E. Lacroix, F. Nashashibi, C. Laurgeau, Vehicle detection combining gradient analysis and adaboost classification, in: IEEE Conf. on Intelligent Transportation Systems, 2005, pp. 1084–1089.
[4] D. Ponsa, A. Lopez, F. Lumbreras, J. Serrat, T. Graf, 3D vehicle sensor based on monocular vision, in: IEEE Proc. Intelligent Transportation System, 2005, pp. 1096–1101.
[5] T. Liu, N. Zheng, L. Zhao, H. Cheng, Learning based symmetric features selection for vehicle detection, in: IEEE Symp. on Intelligent Vehicles, 2005, pp. 124–129.
[6] D. Ponsa, A. Lopez, Cascade of classifiers for vehicle detection, in: LNCS, Springer, Berlin, 2007, pp. 980–989.
[7] A. Giachetti, M. Campini, V. Torre, The use of optical flow for road navigation, IEEE Trans. Robot. Autom. 14 (1) (1998) 34–38.
[8] P.H. Batavia, D.A. Pomerleau, C.E. Thorpe, Overtaking vehicle detection using implicit optical flow, in: IEEE Conf. on Intelligent Transportation System, 1997, pp. 729–734.
[9] H.Y. Chang, C.M. Fu, C.L. Huang, Real-time vision-based preceding vehicle tracking and recognition, in: IEEE Intelligent Vehicles Symposium, 2005, pp. 514–519.
[10] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Prentice-Hall, 2002.
[11] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. SMC-9 (1) (1979) 62–66.
[12] A. Singh, Optical Flow Computation: A Unified Perspective, IEEE Computer Society Press, 1992.
[13] G.P. Stein, O. Mano, A. Shashua, Vision-based ACC with a single camera: bounds on range and range rate accuracy, in: Proc. IEEE Int'l Conf. on Intelligent Vehicles Symposium, 2003, pp. 120–125.
[14] B.F. Wu, C.J. Chen, C.C. Kao, C.W. Chang, S.T. Chiu, Embedded weather adaptive lane and vehicle detection system, in: IEEE Int'l Symposium on Industrial Electronics, 2008, pp. 1255–1260.