Forward Collision Warning System Based on Vehicle Detection and Tracking

Zhipeng Di
Institute of Embedded Software and Systems
Beijing University of Technology
Beijing, China
e-mail: dirischlecht@outlook.com

Dongzhi He
Institute of Embedded Software and Systems
Beijing University of Technology
Beijing, China
e-mail: victor@bjut.edu.cn

***Abstract*—Recently, researches about Forward Collision Warning (FCW) technology has become one of the main parts of the development of Advanced Driver Assistance Systems (ADAS) to increase the road safety and develop autonomous vehicles. This paper presents a research about FCW based on monocular vision, which consists of two processes: vehicle detection and vehicle tracking. In the vehicle detection process, the vertical and horizontal edges and the shadows under the vehicles are combined at first to roughly generate hypothesis of all the vehicles' positions, and then HOG transform and an improved Adaboost classifier are applied to these positions to eliminate non-vehicles. In the vehicle tracking process, frames are handled by an innovative feature point matching method based on Harris detector and normalized cross correlation, and the distance is measured by checking the bottom line of newly tracked vehicles' bounding box. The presented method is verified by experiments, and is proved to have good performance.**

Keywords-intelligent systems; image recognition; feature extraction; vehicle safety;

I. INTRODUCTION

As the increasing demand of road safety nowadays, Advanced Driver Assistance Systems (ADAS) has become a hotspot in relevant studies. ADAS is an integrated system which consists of various positive and negative sensors mounted on the vehicle. These sensors collect information around the vehicle to recognize, detect or track static or dynamic objects that may pose a threat to the driver, and alarm the driver instantly when emergent situations occur, thus avoid traffic accidents effectively. The central part of this field is the detection and tracking of vehicles on road. In relevant studies, cameras, as a representative kind of negative sensor, are widely applied [1]. However, big challenges still exist in vision based studies: The driving environment is semi-structured, which indicates that except for the road surface and the horizon, everything else that may exist can't be exploit as a direct resource to model the environment; Vehicles on the road seen by the camera can have various appearance, in the aspect of shapes, sizes, colors, etc; Vehicle's relative position to the camera is not fixed: Other vehicles can appear right in front of the windshield or partially appear from the lateral side and even overlap each other. As a result, most similar researches are limited to strict conditions. Their experiments are carried out on urban roads where the velocity of vehicle is generally lower than 50km/h. Moreover, good performance can only be achieved in good weather, while complex installation and calibration of the camera is a must.

In this paper, in order to tackle with the aforementioned problem, and develop an effective and robust FCW system, we devise a method which combines vehicle detection and vehicle track process together to produce reliable results using a monocular camera mounted in the car. The rest of the paper is organized as follows: Section II presents the vehicle detection process which finds all reliable positions of vehicles in a given frame. Section III presents vehicle tracking process and distance measuring method, as the preparative step to remind drivers when the distance is too close. Section IV reports the experiment results. Section V concludes this paper.

II. VEHICLE DETECTION

Searching every pixel in every frame in the video stream to locate all possible positions of the vehicles is too time consuming to put into practice. As a general rule, we divide the whole process into two steps: hypothesis generation and hypothesis verification. In the former step, we use edges and shadows as clues to roughly determine the borders of the vehicles in the image. The results are rather coarse that can produce some sub-images of non-vehicles. In the latter step, all the sub-images are put into an Adaboost classifier which is trained to separate vehicles from non-vehicles.

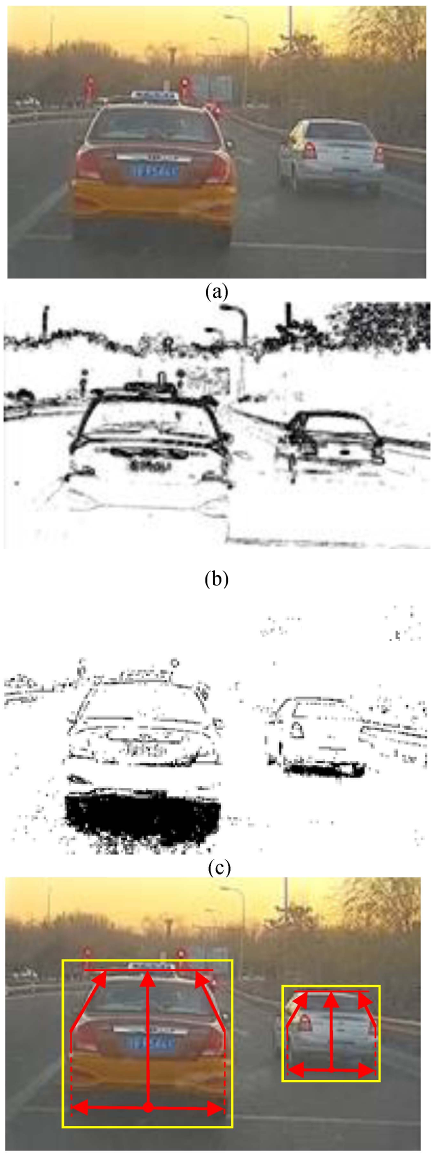
A. Hypothesis Generation

The reason why edges and shadows are used as clues is that they are notable features of the appearance of vehicles. No matter what looking the vehicle has, the edges of the car's trunk, bumper, taillight, windshield, will always exist and be convenient for extraction. That's also true for the shadow underneath the vehicle, when driving in the daylight. They also have clear representations in terms of image intensity, and can be handled quickly through simple arithmetic calculations to get a preliminary result.

Shadow underneath the vehicle has some satisfying characteristics for being a practical feature. Although it is influenced by the intensity of ambient light, the space between the chassis and the ground is so small that only a tiny amount of diffusion light from the lateral side, rather than the direct light, can have an impact on the shadow [2]. Consequently, the section at the bottom of the vehicle will always look darker than other parts of the road. In order to

978-1-5090-0880-3/16/$31.00 ©2016 IEEE

find the shadow, a threshold must be determined to binarize the whole image. The region of interest (ROI) that we select is the area in front of our car within 15 meters, and has the same width as our car. The position of this region is constant in the view seen from the front windshield when the camera is fixed behind the rear-view mirror. The histogram of pixels in this region is collected, whose distribution contains two hops from the two sides of the shadow area, thus pixels between the two hops is located in the shadow. The fluctuation of the mean value is virtually limited to 50 units, showing a good stability on various light conditions. It's reasonable to pick the median as the threshold of binaryzation. The binary image is showed in Fig. 1(b), and it can be seen that shadows are nicely extracted.



(a) (b) (c) (d)

Figure 1. (a) Original (b) Binary image (c) Edge image (d) Bounding box.

Edges are usually extracted by applying convolutions of gradient operator to the raw image entirely. This operation is so common that fast algorithms like Sobel filter or Laplace filter are implemented on various hardware platforms [3]. However, calculation of image gradient is vulnerable to noise generated from various sources. To improve the stability of the extraction of edges and weaken the influence of the light, we propose a method which binarizes the image based on the local mean value of the pixel intensity after apply Sobel operator to the image. We add γ (between 0.4 and 0.5 in this paper) parameter to control the sensitivity of the edge response when gradient is calculated, so that subtle changes of illumination can be reflected more precisely, as show in (1), where $f_I(p)$ is the value after applying our method. Other parameters are: $G_I(p)$ is the original gradient value, Δ is the neighborhood of the center pixel, n is the number of pixels in the neighborhood (usually takes 8). Edge image is shown in Fig. 1(c).

The last step of hypotheses generation is to establish the bounding box of every possible car. To ensure the accuracy of the result, a proper starting point of detection is necessary. The center locations of all the shadow areas are chosen for this purpose, which can avoid the deformation near the tire. The edge image and the shadow image is combined to determine the baseline of the vehicle, and then the detection process is carried out from the starting point along the vertical edge on both side until the horizontal edge on the top is reached. A proper width-height ratio (between 0.8 and 1.3) is set in advance to keep the search in a reasonable range. The bounding box determined by our method is shown in Fig. 1(d).

$$f_I(\mathrm{p}) = \begin{cases} 1 & |G_I(p)| > \frac{\gamma}{\mathrm{n}} \sum_{p \in \Delta} f(p) \\ 0 & others \end{cases} \qquad (1)$$

B. Hypotheses Verification

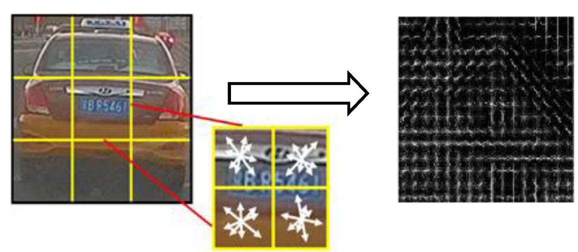


Figure 2. HOG feature extraction.

Vehicle verification is essential because it filters out sub-images of non-vehicles proposed by the hypotheses generation step [4]. So the feature picked for describing images should have a strong ability to distinguish two images. The feature that we use is Histogram of Oriented Gradient (HOG), because it has several advantages that make it suitable for describing vehicles: Quantified steps in every direction accumulate their own gradient value separately, offsetting the influence cause by irrelevant movement such as shift or zooming. The histogram of every local sector is normalized to one, offsetting the influence caused by change of light condition [5]. In this paper, we scale the vehicle's image into 64x64 pixels resolution, and then slice it into 9 blocks. Every block is then divided into 4 cells, in which collection of the histogram proceeds towards 20 directions that are evenly distributed, as shown in Fig. 2.

After every eighty-dimensional feature is obtained, a classifier is needed to separate the features points of vehicles and non-vehicles. To form a nice decision boundary that is moderately learned from the provided samples, we use an improved Adaboost classifier which proves to be satisfactory in performance. The weighted parameter of weak classifiers is altered as (2) to boost the accuracy of classification. During the *i*th iteration, the sum of weight that belongs to every weak classifier p_i appears in the exponential term, leading to a rapid increase when the combination of the new weak classifiers makes a decision with a small error rate ε_i. Therefore, weak classifier which has a better ability to recognize positive examples gets a greater value, and the final strong classifier gets a lower false accept rate (FAR).

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right) + k * e^{p^i} \tag{2}$$

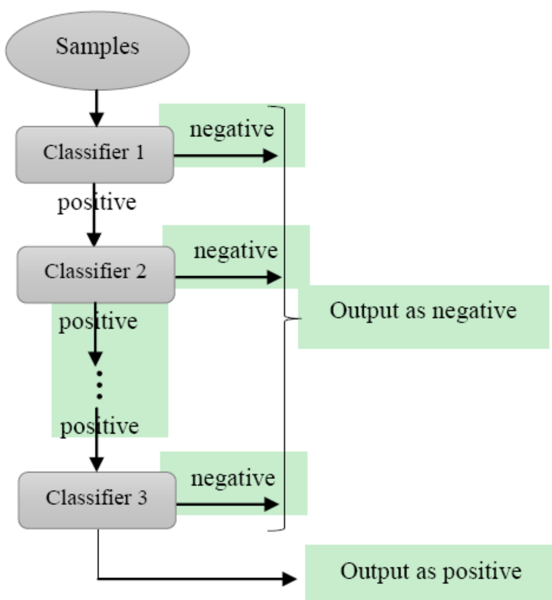


Figure 3. Structure of cascading adaboost.

To take advantage of the superiority of one single classifier which has a low FAR, we adopt a cascading structure as shown in Fig. 3. When samples are put into any of the sub-classifiers, only positive ones can pass and enter the next sub-classifier or be output as positive if the current sub-classifier is the last one, whereas negative ones are output as negative directly without further classification. The first few levels of sub-classifier have simple structures, and are trained by samples with fewer features. Because of the low FAR of every sub-classifier, it's very difficult for negative samples to pass through even a few upstream sub-classifiers, let alone the last ones. This characteristic guarantees the simplicity of computation without losing accuracy. We make the number of sub-classifiers equal to 3, preventing excessive usage of classifiers and wasting resources. As sample input, the HOG features are organized as follows: Before the first sub-classifier, feature values in every four adjacent directions are merge into one so that the amount of features in each block shrinks from 80 to 20; Similarly, before the second sub-classifier, merging happens in every two adjacent directions. Finally, the whole 80 features are put into the last sub-classifier without any change. The class labels obtained from this classifier have a very high probability to be consistent with real situations.

III. VEHICLE TRACKING

After vehicle detection follows the vehicle tracking process. The basic duty of vehicle tracking is to focus on the target all long until it's out of the range of the camera's eyesight. The cost of tracking detected target is much cheaper than carrying out another detection process in subsequent frames, because it exploits the property of continuity between adjacent frames and only tackles known local areas rather than the whole image [7].

Feature points matching is a practical way to achieve this goal, such as Harris, SIFT and Shi-Tomasi [8]. We use an improved Harris algorithm which detects corners with various scales, thus has a higher precision. The scales information is obtained from the Gaussian space which is transformed from the original image using Gaussian kernel. We predefine a set of scale value which consists of 1 to k times of a base scale σ_0 and calculate the second moment of the Harris operator as (3), noted as M. The response function of traditional Harris detector is not only dependent on M but also on an empirical parameter k, which is usually between 0.04~0.06. Often, it's almost impossible to find an optimal value of k to minimize its introduced error. Consequently, we relate the respond function with the predefined scale value set to let M be independent of parameter k, as shown in (4). The term ε (usually takes 10^{-6}) is a very small value to prevent the denominator from reaching zero under extreme circumstances. This method is applied in all the existing sub-images within their bounding box, and all pixels which have a high respond value (greater than 30 in our experiment) are considered as feature points.

$$\mathrm{M}(\mathrm{x},\mathrm{y},\sigma_i) = G(x,y,\sigma_i) \otimes \begin{bmatrix} g_x^2 g_x g_y \\ g_y g_x g_y^2 \end{bmatrix} \tag{3}$$

$$\mathrm{CRF}(\mathrm{x},\mathrm{y},\sigma_i) = \frac{\det(M(\sigma_i))}{tr(M(\sigma_i))+\varepsilon} \tag{4}$$

Extraction of feature points takes place in every frame. Between two adjacent frames, feature points are matched to each other according to the normalized cross correlation (RCC) criterion, as show in Fig. 4. Every point looks for the point to match only in its neighborhood, avoiding unnecessary searching cost. The range of RCC is between -1 and 1, while -1 means low correlation and 1 means high correlation. Any pixel pair whose RCC is greater than 0.9 is considered as matched pairs, and produces a motion vector to evaluate the vehicle's motion as time elapses. If more than half of the feature points are found and matched, the vehicle is successfully tracked, and its bounding box is generated at its new location. The new distance is given directly by the bottom line of the bounding box because its vertical position in the view has a fixed relation with the distance to measure, assuming that the ground is flat. Otherwise, the vehicle is lost and all its feature points are abandoned without further reference. Last but not least, vehicle detection needs to be re-invoked after a given time interval (every 24 frames in this paper) over and over in order to find new target which newly comes into sight or reappears again after it's out of tracking.

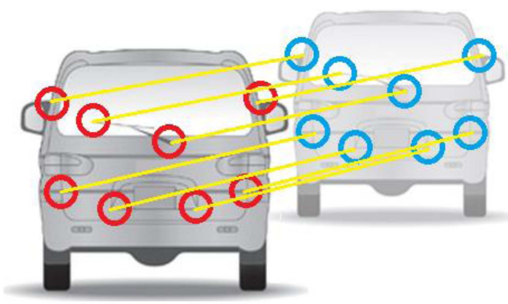


Figure 4. Feature point matching.

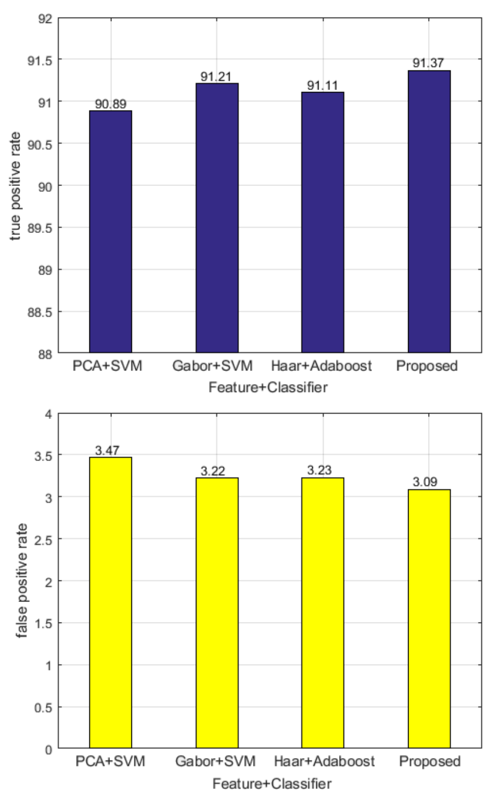


Figure 5. Comparesion of t_p and f_p with other method.

IV. Experiment Results

To verify the effectiveness of our method, we implement it in C++ using OpenCV library on 32-bit windows. The hardware platform is a laptop, running with an Intel® Core™ i5 3.20-GHz CPU and 4.00GB RAM. The videos prepared for testing is recorded through an automobile event data recorder whose frame rate is 24 fps, and their resolution is adjusted to 300x400 to reduce the workload of image processing.

Firstly, to evaluate the performance of the vehicle detection part, adequate amount of vehicle and non-vehicle samples under various conditions (normal, low light, partially visible) are fetched from our video and chosen as input, and the results are compared to three state-of-the-art algorithms: PCA combined with SVM in [9], Gabor Wavelet combined with support vector machine (SVM) in [10], Haar-like Wavelet combined with Adaboost in [11]. Two values are taken into consideration: true positive rate (t_p) and false positive rate (f_p) of the detected results. The results are shown in Fig. 5, from which advantages in detection accuracy of the proposed method over other existing methods can be seen.

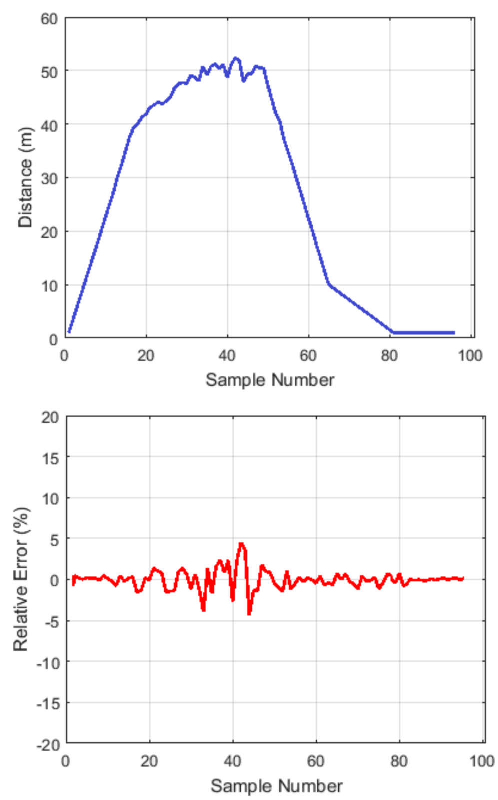


Figure 6. Measured distance and its relative error.

Next, to evaluate the accuracy of the vehicle tracking and distance measurement, a car right in front of our car is prepared as the tracking target, and the distance is measured by our method and compared to the real distance sampled by a laser sensor. The test is carried out in various situations: driving speed can be low or high, while the distance can be near or far. The results are shown in Fig. 6. It can be seen that the error grows notably when the distance is over 40, but the maximum relative error is still under 5%. This is mainly because as the distance increases, feature points become more and more obscure and hard to match with each other, thus the generation of new bounding box is interfered. Fortunately, handling for those faraway vehicles isn't necessarily very urgent, thus errors to this extent are often tolerable. When the distance is near, the accuracy is quite satisfying.

V. Conclusion

Although the bottleneck faced by researches using cameras in ADAS systems is the strong demand of computing power for video processing, the IC industry are making great progress in the development of fast video processors which makes complex algorithms about vehicle detection and vehicle tracking possible [12]. Our paper presents a forward collision warning system that can detect other vehicles appear in front of the car on road and track them, measure their distance to warn the driver when the distance is too close which can potentially cause an accident. Experiment result have shown that our detection method has a high detection rate, and our tracking and distance measuring method has a high accuracy in real environment.

In spite of the achievements of this paper, there are still some future works to be done to make the whole system more applicable to real circumstances. For instance, the assumption that the ground is flat, which simplifies the relation between the bounding box's bottom line and the vehicle's distance, does not always coincide with the reality. Further analysis of road environment is still needed, in order to make the results more precise. Moreover, some of the algorithms we used, such as cascading Adaboost, are still too computationally expensive to be deployed on embedded platforms, so minimized and fast-speed version of these algorithms are expected to be studied. Last but not least, exploration of high level semantic information such as vehicle behavior could be beneficial, and will pave the way for more sophisticated functions concerned with the possible future of road traffic, namely automated driving.

Acknowledgment

This work is supported by PXM2015_014204_500211 projects of Science and Technology Innovation Platform in Beijing City, China

References

[1] M. Nieto, O. Otaegui, G. Velez, J.D. Ortega, A. Cortes, "On creating vision-based advanced driver assistance systems," Iet Intell Transp Sy, 2015, pp. 59-66.

[2] S. Sivaraman, M.M. Trivedi, "Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis," Ieee T Intell Transp, 2013, pp. 1773-1795.

[3] Z.H. Sun, G. Bebis, R. Miller, "On-road vehicle detection: a review," Ieee T Pattern Anal, 2006, pp. 694-711.

[4] M.I. Arenado, J.M.P. Oria, C. Torre-Ferrero, L.A. Renteria, "Monovision-based vehicle detection,distance and relative speed measurement in urban traffic," Iet Intell Transp Sy, 2014, pp. 655-664.

[5] T. Yu, H. Shin, "Detecting partially occluded vehicles with geometric and likelihood reasoning," Iet Comput Vis, 2015, pp. 174-183.

[6] J.M. Guo, H. Prasetyo, K. Wong, "Vehicle verification using Gabor filter magnitude with Gamma distribution modeling," Ieee Signal Proc Let, 2014, pp. 600-604.

[7] C.H. Shen, J. Kim, H.Z. Wang, "Generalized kernel-based visual tracking," Ieee T Circ SystVid, 2010, pp. 119-130.

[8] A. Broggi, E. Cardarelli, S. Cattani, P. Medici, M. Sabbatelli, "Vehicle detection for autonomous parking using a soft-cascade AdaBoost classifier," 2014 Ieee Intelligent Vehicles Symposium Proceedings, 2014, pp. 918-923

[9] J. Wu, X. Zhang, "A PCA classifier and its application in vehicle detection," Proceedings of the IEEE International Joint Conference on Neural Networks, vol. 1, 2001, pp. 600–604.

[10] Z.H. Sun, G. Bebis, R. Miller, "On-road vehicle detection using Gabor filters and support vector machines," International Conference on Digital Signal Processing, 2002, pp. 1019–1022.

[11] R. Lienhart, A. Kuranov, V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," Proceedings of the 25th German Pattern Recognition Symposium, 2003, pp. 297–304.

[12] H. Hassannejad, P. Medici, E. Cardarelli, P. Cerri, "Detection of moving objects in: roundabouts based on a monocular system," Expert Syst Appl, 2015, pp. 4167-4176.