

# 미약한 시각 특징과 Haar 유사 특징들의 강화 연결에 의한 도로 상의 실 시간 차량 검출

## Real Time On-Road Vehicle Detection with Low-Level Visual Features and Boosted Cascade of Haar-Like Features

삼 아디카리, 유 현 중, 김 형 석\*  
(Shyam Prasad Adhikari<sup>1</sup>, Hyeon-joong Yoo<sup>2</sup>, and Hyongsuk Kim<sup>1</sup>)

<sup>1</sup>Chonbuk National University

<sup>2</sup>Sangmyung University

**Abstract:** This paper presents a real-time detection of on-road succeeding vehicles based on low level edge features and a boosted cascade of Haar-like features. At first, the candidate vehicle location in an image is found by low level horizontal edge and symmetry characteristic of vehicle. Then a boosted cascade of the Haar-like features is applied to the initial hypothesized vehicle location to extract the refined vehicle location. The initial hypothesis generation using simple edge features speeds up the whole detection process and the application of a trained cascade on the hypothesized location increases the accuracy of the detection process. Experimental results on real world road scenario with processing speed of up to 27 frames per second for 720x480 pixel images are presented.

**Keywords:** vehicle detection, shadow, symmetry, haar-like features

### I. INTRODUCTION

Increasing safety concerns on the road have led to the investigation and deployment of many IVAS (Intelligent Vehicle Assistant Systems) in vehicles. A fast and reliable succeeding and preceding vehicle detection is one of the critical tasks of IVAS. Vision sensors have been widely used in the literatures for vehicle detection [1-4].

A method to detect vehicles based on the shadow underneath the vehicle is presented in [1]. Symmetry as a cue for vehicle detection has been studied in [2] where a 'symmetry finder' is used to find mirror symmetry based on intensity. Vertical and horizontal edges as strong cue for vehicle detection have been studied in [3]. The left and right position of the vehicle is found using the vertical profile of the edge image and shadow is used to find the bottom of the vehicle. A multiscale approach that combines sub-sampling with smoothing is presented in [4], where possible vehicle location is computed from the maxima and minima of horizontal and vertical edge profiles. Recently Haar-like feature detectors have been used widely for object detection [5-8]. Viola and Jones [7] used the Haar-like features and Adaboost for face detection. A pedestrian detection system integrating image intensity with motion information in Haar-like features is presented in [8]. A motion and Haar-like feature detector to detect vehicle is proposed in [6].

The use of the low level image features alone is not robust for effective detection and the use of classifiers alone for detection is

computationally expensive due to the scanning nature of the detectors. These methods individually implemented are not suitable for real-time applications. To overcome the difficulties of these approaches we propose a method combining the two approaches so that the final detector is robust and efficient for real-time application. In this paper, we propose a method combining the low level image features and the Haar-like feature detector for increased performance of the detector. The initial hypothesis of the vehicle is generated using computationally cheap edge features which speeds up the detection process and a trained cascade of Haar-like features is applied on the hypothesized location which increases the robustness of the detection process.

The rest of the paper is organized as follows; Section II describes relevant background theory and the proposed method, section III presents the experimental results and section IV presents the conclusions.

### II. PROPOSED METHOD

The method proposed in this paper consists of two stages. In the first stage, the candidate vehicle location in an image is found using the low-level horizontal edge features and symmetry. Based on the horizontal edges, symmetry and local perspective constraints, ROI for vehicle search is defined. In the second stage a cascade of classifiers trained on vehicles using Haar-like features and Adaboost [9] is applied to the ROI to detect the vehicles.

#### 1. Initial vehicle hypothesis generation

Vehicles exhibit prominent horizontal edges and a shadow region is present underneath the vehicles. These two features characteristic of vehicles are used to estimate the position of the vehicle on the road. Horizontal Sobel mask is used to extract the horizontal edges and a weighted edge profile is drawn. The

\* 책임저자(Corresponding Author)

논문접수: 2010. 11. 25., 수정: 2010. 11. 26., 채택확정: 2010. 12. 23.

삼 아디카리, 김형석: 전북대학교 전자공학부

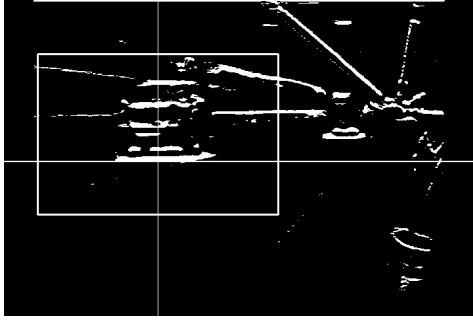
(all.shyam@gmail.com/hskim@jbnu.ac.kr)

유현중: 상명대학교 정보통신 공학과(yoohj@smu.ac.kr)

※ 상기 논문은 제어·로봇·시스템학회 전북제주시부와 전남 광주 지부와의 합동학술대회에서 초안이 발표되었음.



(a) The original image (720x480).



(b) The vehicle bottom line (horizontal line), symmetry axis (vertical line) and the ROI generated (bounding box) superimposed in the horizontal edge image of (a).

그림 1. 초기의 임시 특징들의 생성.

Fig. 1. Initial Hypothesis generation.

position of maximum in the weighted edge profile [10] gives the initial estimate of the bottom position of the vehicle, Fig. 1.

Vehicles observed from the rear or front are generally symmetric. A fast symmetry detection algorithm [11] suitable for real time application is used to extract the vertical symmetry axis. The horizontal edge pixels are utilized to extract the vertical symmetry axis of the vehicle. The maximum value in the symmetry histogram resulting from the algorithm gives the position of the vertical symmetry axis, Fig. 1. The initial estimate of the vehicle bottom and the symmetry axis along with some local perspective constraints allow to extract a ROI, Fig. 1, for the final search of the vehicle.

## 2. Haar-like features

Haar-like features, Fig. 2, are simple rectangular filters. The Haar-like features have scalar values that represent the difference in the sum of intensities between the adjacent rectangular regions. The feature values can be computed rapidly using integral image representation.

To capture the ad-hoc knowledge about the domain, these features are evaluated at different positions and with different sizes, exhaustively according to the base resolution of the classifier. For example, when the classifier resolution is 24x18 pixels, 91620 features are generated from the features in Fig. 2. Each feature is evaluated on all the training samples and one-dimensional probability density for each of the object and non-object class is calculated as shown in Fig. 3. A threshold that separates these two distributions is selected for each feature. These features along with their respective thresholds and polarity form the weak classifiers for the learning algorithm.



그림 2. Haar 유사 특징들의 예.

Fig. 2. Examples of the Haar-like features set.

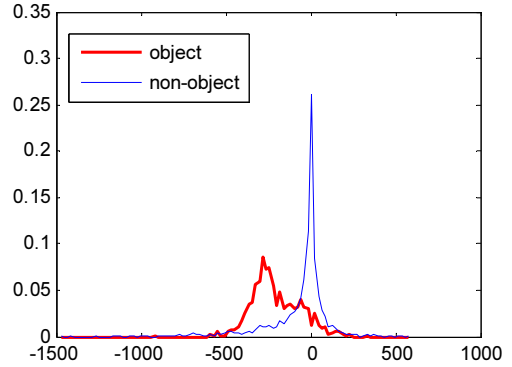


그림 3. 특징 값 분포의 예.

Fig. 3. Example of a feature value distribution.

A weak classifier can be mathematically described as,

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) > p\theta \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

where  $x$  is the base resolution of the classifier, the Haar-like feature, the threshold for the feature and the polarity indicating the direction of inequality.

## 3. Adaboost based learning

Given a feature set and a training set of positive and negative images, any machine learning approach can be used to learn a classification function. In our system a variant of Adaptive Boosting (Adaboost) is used. Adaboost is a machine learning algorithm which is used to boost the classification performance of a simple learning algorithm. Here, the Haar-like features, Eq. (1), are considered as weak learners. Adaboost is used to select the Haar-like features and to train the classifier. A small number of discriminative weak classifier is selected by updating the sample distribution. The prediction of the strong classifier is produced through a weighted voting of the weak classifiers. The pseudo code of a variant of Adaboost used in the implementation is given in Fig. 4.

A cascade of the classifiers is constructed to act as a rejection cascade to increase the detection performance and reduce the computation time [5]. The key idea behind the cascade is to use the initial simple classifier stages to reject the majority of the sub-windows before more complex classifiers are called upon to achieve low false positives. Each stage in the cascade is trained to achieve very high detection rate and low false positive rate (<0.5). The schematic of the cascade is shown in Fig. 5. A positive result from the first classifier triggers the evaluation of the second classifier and a positive result from the second triggers the third stage. A negative outcome at any point leads to the immediate rejection of the sub window. Since, in any single image an overwhelming majority of the windows are negative, the cascade attempts to reject as many negatives as possible at the earliest stage possible.

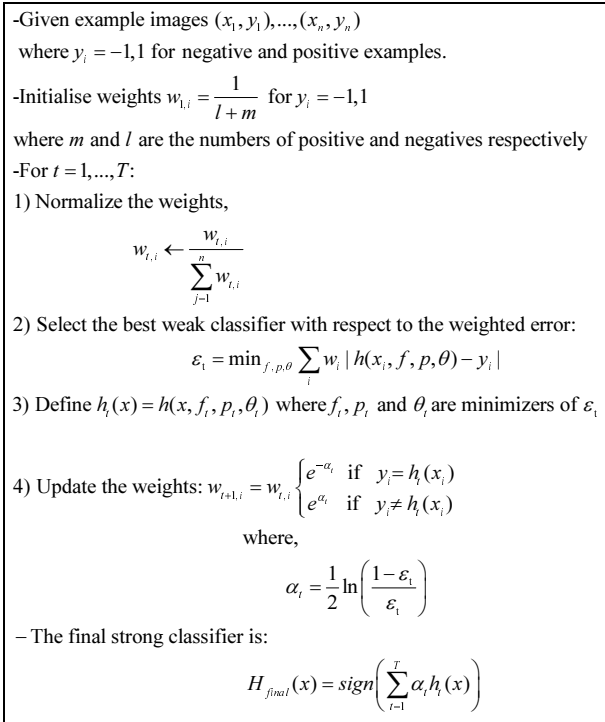


그림 4. 이산 adaboost 알고리즘의 pseudocode.  
 Fig. 4. Pseudocode for discrete adaboost.

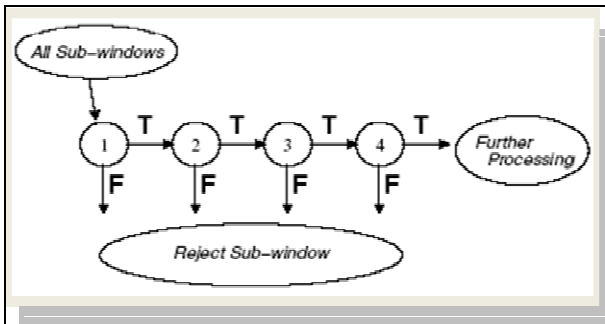


그림 5. 순차적 검출 모형.  
 Fig. 5. Schematic depiction of the detection cascade.

4. Scanning the detector

In the detection phase, the trained cascade is scanned across the image at multiple scales and locations. The scaling can be achieved by scaling the feature rather than scaling the image itself. The detector is also scanned across location. Subsequent locations are obtained by shifting the window some number of pixels  $\Delta$ . The choice of  $\Delta$  affects the speed and accuracy of the detector. Though a suitable value of the scale and  $\Delta$  helps to optimize the detection process, the detector still has to process millions of windows in a single image. This process is very time consuming and is not suitable for real-time applications.

Instead of applying the cascade detector at all scales and locations across an image, we propose to apply the detector only in the ROI generated using the computationally cheap low level visual features. The number of windows to be scanned in the ROI is greatly reduced in comparison to scanning the whole image, Fig. 6. This increases the overall speed of the detector and the proposed algorithm can be implemented in real time.



그림 6. 그림 1(b)의 ROI에 적용된 연쇄적 분류기.  
 Fig. 6. Cascade classifier applied to the ROI generated in Fig. 1(b).

III. EXPERIMENTAL RESULTS

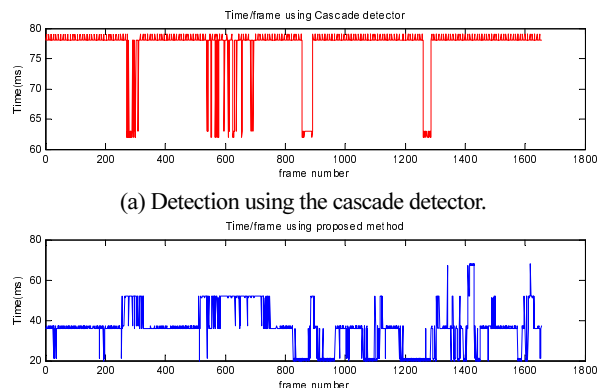
This section describes the final vehicle detection system. The proposed algorithm is implemented in Visual C++ and OpenCV 1.0 on a 2.6 GHz computer.

1. Training the detector cascade

The vehicle training set consisted of 500 hand labeled vehicles and aligned to a base resolution of 24x18 pixels, Fig. 7. The vehicles were extracted from images of a video taken from a camera mounted on the side view mirror of a host car while driving in a natural urban traffic. The final detector is a 20 layer cascade with a total of 227 features.



그림 7. 학습에 사용된 차량의 앞면 영상 예.  
 Fig. 7. Example of the front view of car used for training.



(b) Detection using the proposed method. The processing time for each frame using the proposed method is always less than the time required using the cascade detector alone.

그림 8. 프레임 당의 처리 시간.  
 Fig. 8. Processing time per frame.

## 2. Experiments on real-world test set

The algorithm was tested on a separate video sequence taken with the same arrangement of the camera. This set consisted of a total of 1654 frames of resolution 720x480 pixels. Two separate experiments were carried out; one using the proposed algorithm and the other using boosted cascade of Haar-like features only. The cascade detector was scanned with an initial scale of 1.2 and  $\Delta=1$ . The proposed method required an average of 37 ms to process a frame giving a frame rate of 27 frames per second. The detector using the cascade classifier alone required an average of 77 ms to process a frame giving a frame rate of 13 frames per second, Fig. 8.

The variation in the processing speed of the detector using the cascade detector alone is due to the different context of the image. Some sub windows need the evaluation of more number of stages for effective discrimination whereas some require few stages. The increase in the speed of the proposed detector as seen from Fig. 8 is due to the first stage of ROI extraction using the computationally cheap low level edge features. The ROI restricts the search space for the cascade detector and the number of sub

windows to be evaluated is reduced. The variation in the processing time of the proposed detector is due to two factors. Symmetry detection process depends upon the number of horizontal pixels present on the image, thus the symmetry extraction time for different frame varies. The processing time also depends on the size of the extracted ROI. Some of the results of the proposed algorithm are presented in Fig. 9.

## IV. CONCLUSIONS

A two stage real time vehicle detection system is proposed. The first stage uses low-level horizontal edge features and symmetry to rapidly compute the candidate vehicle positions in an image. A ROI formed by using the features and local perspective constrains provides the final search space where Haar-like feature based detector is applied to detect the vehicle. The ROI found using computationally cheap visual features, helps to restrict the search space for the computationally intensive Haar-like feature detector. The experimental results show that the proposed method increases the speed of the detector significantly, making it suitable for real time application.

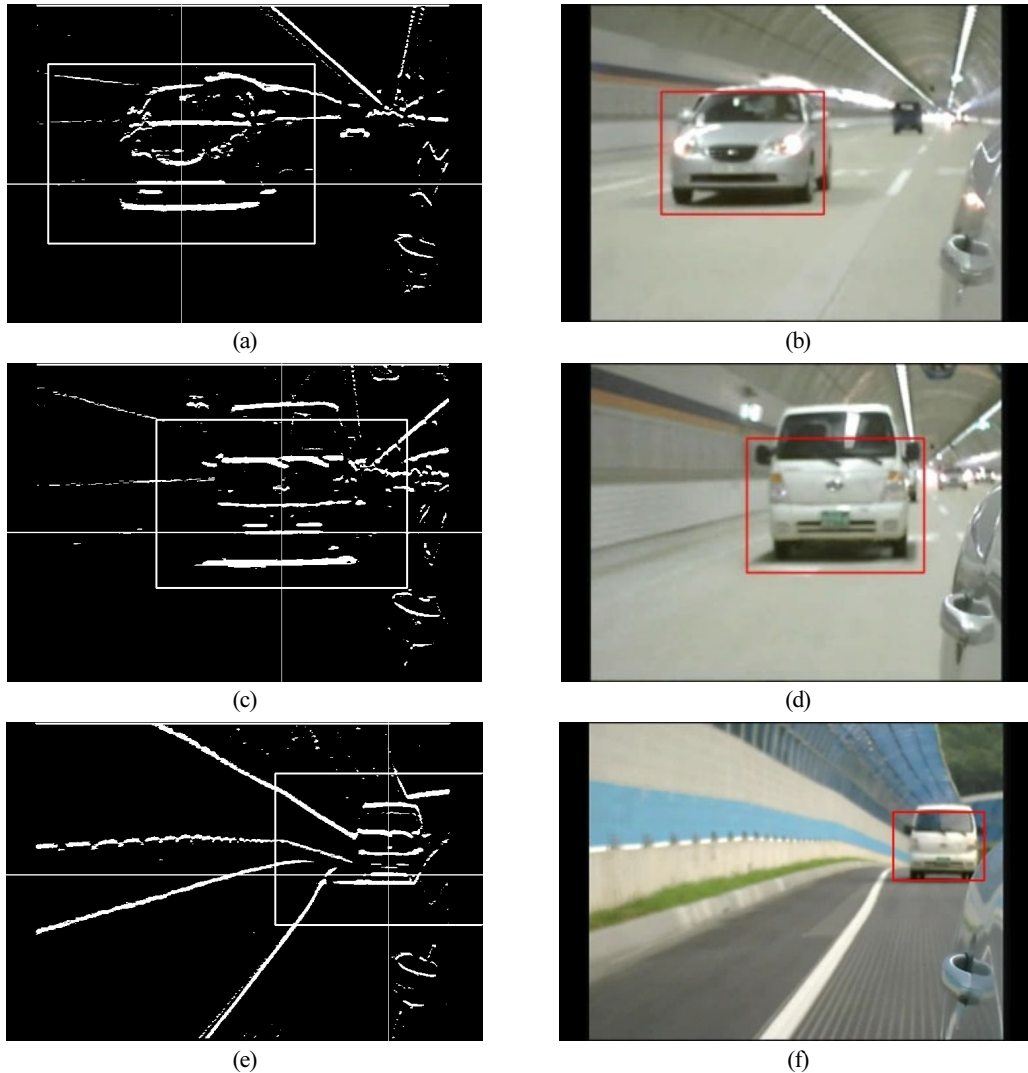


그림 9. 차량 검출; (a, c, e) 저면 선, 대칭축가 둘러 박스와 겹쳐 그린을 에지 영상; (b, d, f) 검출된 차량.

Fig. 9. Vehicle detection; (a, c, e) the edge image with the bottom line, symmetry axis and the bounding box superimposed; (b, d, f) the detected vehicles respectively.

## REFERENCES

- [1] C. Tzomakas and W. Sleen, "Vehicle detection in traffic scenes using shadows," Technical Report 98-06, Institut fur Neuroinformatik, Ruth-Universitat, Bochum, Germany, 1998.
- [2] T. Zielke, M. Brauckmann, and W. von Seelen, "Intensity and edge based symmetry detection with application to car-following," *CVGIP:Image Understanding*, vol. 58, pp. 177-190, 1993.
- [3] N. Matthews, P. E. An, D. Charnley, and C. J. Harris, "Vehicle detection and recognition in greyscale imagery," *Control Eng. Practice*, vol. 4, pp. 473-479, 1996.
- [4] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time pre-crash vehicle detection system," *Proc. of IEEE Int'l Workshop Application of Computer Vision*, Dec. 2002.
- [5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
- [6] H. Bai, J. Wu, and C. Liu, "Motion and Haar-like feature based vehicle detection," *Proc. of the 12th International Multi-Media Modelling Conference*, pp. 356-359, 2006.
- [7] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, 2004.
- [8] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Proc. of Ninth IEEE International Conference on Computer Vision*, 2003.
- [9] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, pp. 297-336, 1999.
- [10] S. P. Adhikari, H. Cho, H. Yoo, C. Yang, and H. Kim, "On-road succeeding vehicle detection using characteristic visual features," *Trans. on KIEE*, vol. 59, no. 3, 2010.
- [11] W. H. Li, A. M. Zhang, and L. Kleeman, "Bilateral symmetry detection for real-time robotics applications," *The International Journal of Robotics Research*, vol. 27, no. 7, pp. 785-814, 2008.



## 삼 아디카리

2005년 National Institute of Technology, India(공학사). 2008년 National Institute of Technology, India(공학석사). 2009년~현재 전북대학교 전자정보공학부 박사과정. 관심분야는 멀티미디어와 통신 시스템, 로봇비전, 원격제어.



## 유 현 중

1982년 서강대학교 전자공학과(공학사). 1991년 미조리대학교 전기및컴퓨터공학과(공학석사). 1996년 미조리대학교 전기및컴퓨터공학과(공학박사). 1996년~현재 상명대학교 정보통신학과 교수. 관심분야는 패턴인식, 컴퓨터비전.



## 김 형 석

1980년 한양대학교 전자공학과(공학사). 1982년 전북대학교 전기공학과(공학석사). 1992년 University of Missouri, Columbia(공학박사). 1993년~현재 전북대학교 전자정보공학부 교수. 관심분야는 로봇비전, 로봇센서시스템, 아날로그

병렬처리 회로.