IEEE ISIE 2005, June 20-23, 2005, Dubrovnik, Croatia

Road Vehicle Recognition in Monocular Images

M.A. Sotelo*, J. Nuevo*, L.M. Bergasa*, M. Ocaña*, I. Parra*, D. Fernández*
*University of Alcalá / Department of Electronics, Alcalá de Henares, Madrid, Spain
michael,jnuevo,bergasa,mocana,parra,llorca@depeca.uah.es

Abstract— **This paper describes a monocular vision-based Vehicle Recognition System in which the basic components of road vehicles are first located in the image and then combined with a SVM-based classifier. The challenge is to use a single camera as input. This poses the problem of vehicle detection and recognition in real, cluttered road images. A distributed learning approach is proposed in order to better deal with vehicle variability, illumination conditions, partial occlusions and rotations. The vehicle searching area in the image is constrained to the limits of the lanes, which are determined by the road lane markings. By doing so, the rate of false positive detections is largely decreased. A large database containing thousands of vehicle examples extracted from real road images has been created for learning purposes. We present and discuss the results achieved up to date.**

I. Introduction

This paper describes a monocular vision-based Vehicle Recognition System in the framework of Intelligent Transportation Systems (ITS) technologies. In our approach, the basic components of road vehicles are first located in the image and then combined with a SVM-based classifier. The challenge is to use a single camera as input, in order to achieve a low cost final solution that meets the requirements needed to undertake serial production. A monocular imaging device (a single FireWire digital camera) is deployed to provide "indirect range" measurements using the laws of perspective. Some previous works use available sensing methods such as radar [1], stereo vision [2], or a combination of both [3]. Only a few works deal with the problem of monocular vehicle detection using symmetry and colour features [4] [5], or pattern recognition techniques [6]. Detecting a vehicle in a monocular image poses the general problem of object detection in static images. This is a complex problem as long as it requires that the object class exhibits high interclass and low intraclass variability. In addition, vehicle detection should perform robustly under variable illumination conditions, variable rotated positions, and even if some of the vehicle parts are partially occluded.

Object detection techniques can be classified into three major categories, as described in [7]. The first category is represented by model-based systems in which a model is defined for the object of interest and the system attempts to match the model to different parts of the image in order to find a fit. Unfortunately, road vehicles can be regarded as quite a variable class that makes it impossible to define a model that represents the class in an accurate, general way. In consequence, model-based systems are of little use for vehicle recognition purposes. The second category are image invariance methods which perform a matching based on a set of image pattern features that, supposedly, uniquely determine the object being searched for. Road vehicles do not exhibit any deterministic image pattern relationships because of its large variability (different types of vehicle models depending on manufacturers). For this reason, image invariance methods are not a viable option in order to solve the vehicle recognition problem. The third category of object detection techniques is characterised by example-based learning algorithms. The salient features of a class are learnt by the system based on a set of examples. This type of technique can provide a solution to the vehicle recognition problem as long as the following conditions are met.

- A sufficiently large number of vehicle examples are contained in the database.
- The examples are representative of the vehicle class in terms of variability, illumination conditions, and position and size in the image.

Example-based techniques have been previously used in natural, cluttered environments for pedestrian detection [8] [9]. In general, these techniques are easy to use with objects composed of distinct identifiable parts arranged in a well-defined configuration. This is the case of road vehicles, where a distributed learning approach based on components [7] is more efficient for object recognition in real cluttered environments than holistic approaches [10]. Distributed learning techniques can deal with partial occlusions and are less sensitive to object rotations. However, in spite of their ability to detect objects in real images, we propose to reduce the vehicle searching space in an intelligent manner, based on the road image, so as to increase the performance of the detection module. Accordingly, road lane markings are detected and used as the guidelines that drive the vehicle searching process. The area contained by the limits of the lanes is scanned in order to select candidate regions of interest. These regions are likely to contain the vehicle candidates that are passed on to the vehicle recognition module. This helps reduce the rate of false positive detections. In case that no lane markings are detected, a basic area of interest is used instead covering the front part ahead of the ego-vehicle. The description of the lane marking detection and vehicle recognition systems is provided in the following sections.

II. Candidate Regions of Interest

The system is divided in two modular subsystems. The first subsystem is responsible for lane detection and tracking, as well as lane crossing monitoring. The second subsystem extracts candidate regions of interest by detecting vehicle candidates within the limits established

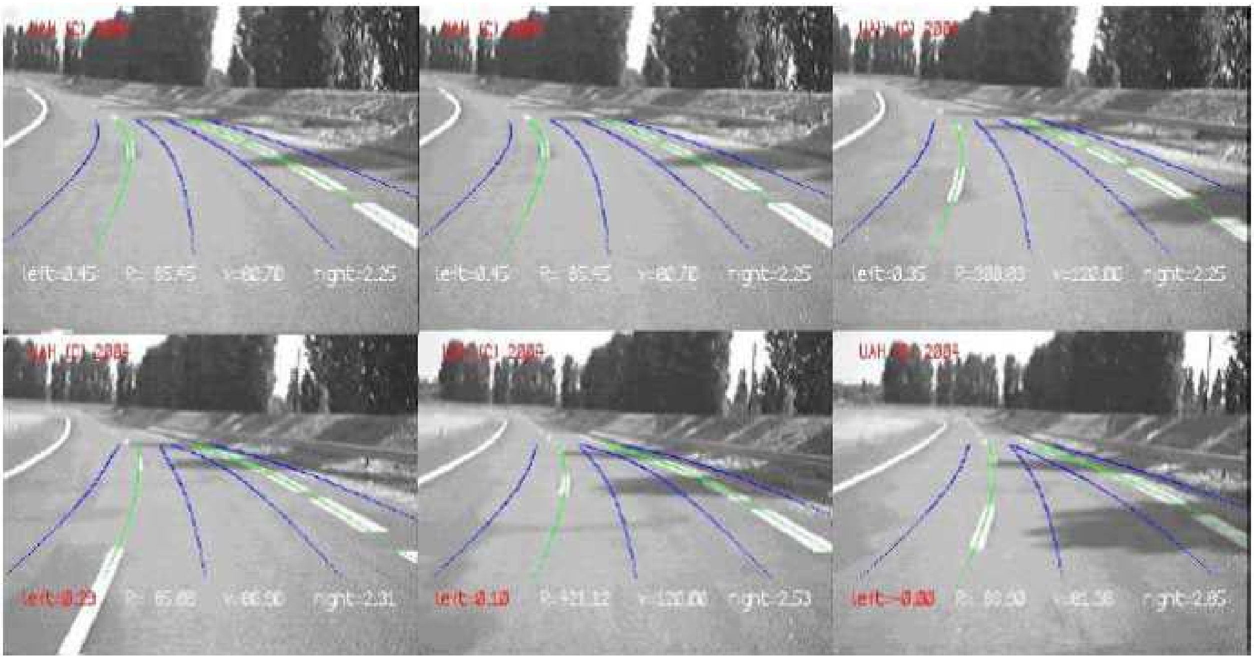

Fig. 1. Lane tracking example in a sequence of images.

by the first subsystem according to the estimated road lanes.

A. Lane detection and tracking

Images obtained from the camera are processed and clothoid curves are fitted to the detected lane markings in order to estimate the road lanes that determine the candidates searching area. The algorithm scans up to 50 lines in the candidates searching area, from 2 meters in front of the camera position to below the horizon. The developed algorithm implements a non-uniform spacing search that reduces certain instabilities in the fitted curve. The final state vector is composed of 6 variables [11] for each line on the road:$c_{oh}, c_{1h}, c_{ov}, c_{1v}, x_o, \theta_o$, where c_{oh} and c_{1h} represent the clothoid horizontal curvature parameters, c_{ov} and c_{1v} stand for the clothoid vertical curvature parameters, while x_o and θ_o are the lateral error and orientation error, respectively, with regard to the centre of the lane. The clothoid curves are then estimated based on lane marking measurements using a Kalman filter for each line. These lines conform the candidates searching area. Figure 1 depicts a sequence of images in which the result of the lane tracking algorithm is overprinted on the road images. The green lines represent the estimated lines of the road, while the blue ones show the lane marking validation area. The example also depicts the error between the left wheel of the car and the left lane (left), the error between the right wheel of the car and the right lane (right), the road curvature radius estimated at a look-ahead distance of 50m (R), and the maximum recommended velocity to bend the curve (V) according to the radius of curvature.

B. Candidate vehicles

An attention mechanism has been devised with the intention of filtering out inappropriate candidate windows based on the lack of distinctive features, such as horizontal edges and symmetrical structures, which are essential characteristics of road vehicles. This has the positive effect of decreasing both the total computation time and the rate of false positive detections. Each road lane is sequentially scanned, from the bottom to the horizon line of the image, looking for collections of horizontal edges that might represent a potential vehicle. The scanned lines are associated in groups of three. For each group, a horizontality coefficient is computed as the ratio of connected horizontal edge points normalized by the size of the area being analysed. The resulting coefficient is used together with a symmetry analysis in order to trigger the attention mechanism. Apart from the detected road lanes, additional virtual lanes have been considered so as to cope with situations in which a vehicle is located between two lanes (for example, if it is performing a change lane manoeuvre). Virtual lanes provide the necessary overlap between lanes, avoiding both misdetections and double detections caused by the two halves of a vehicle being separately detected as two potential vehicles. A virtual lane is located to provide overlap between two adjoining lanes. Figure 2 depicts the candidate regions of interest generated by the attention mechanism in a sequence of images. On average, the system generates 5 candidate windows per frame that are passed on to the classifier. Nonetheless, this figure is bound to change depending on traffic conditions.

III. Vehicle Recognition

The road vehicle class contains quite a large amount of different cars that makes it a non-homogeneous cluster. In

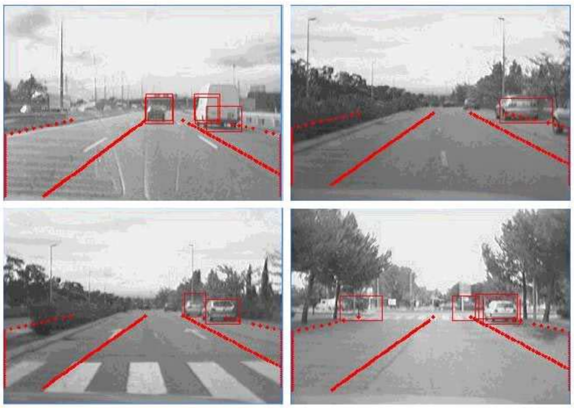

Fig. 2. Generation of candidate regions of interest in a sequence of images.

consequence, it makes sense to use a distributed learning approach in which each individual part of the vehicle is independently learnt by a specialized classifier in a first learning stage. The local parts are then integrated by another classifier in a second learning stage. According to the previous statements, the proposed approach can be regarded as a hierarchical one. By using independent classifiers in a distributed manner the learning process is simplified, as long as a single classifier has to learn individual features of local regions in certain conditions. Otherwise, it would be difficult to attain an acceptable result using a holistic approach. We have considered a total of 3 different sub-regions for each candidate region, as depicted in figure 3. The 3 sub-regions cover the most characteristic parts of the vehicle. Two small sub-regions have been located in the area of the region where the wheels are supposed to be. A third sub-region is located in the central part of the region, covering the area where car plates and rear windshield are usually placed. The locations of the three sub-regions have been chosen in an attempt to detect coherent and structural car features.

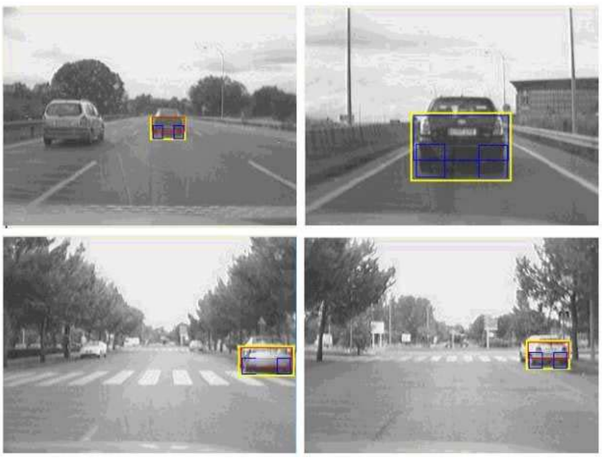

Fig. 3. Decomposition of a candidate region of interest into 3 sub-regions.

A set of features must be extracted from each sub-region and fed to the classifier. Before doing that, the

Fig. 4. Normalized input to the classifi er. Left: original images, Right: Canny images.

entire candidate region of interest is pre-processed using a Canny operator in order to enhance the differential information contained in it (edges). The Canny image provides a good representation of the discriminating features of the car class. On the one hand, edges, both horizontal and vertical, are clearly visible and distinguishable. On the other hand, the vertical symmetry of a car remains unchanged. In addition, edges are not affected by colours or intensity. This property makes the use of edges robust enough to different car models of the same type. In a first attempt, a set of features was extracted from each sub-region using the normalized histogram based on the co-occurrence matrix of the pre-processed sub-region (four co-occurrence matrixes were computed using four different searching vectors). This option was discarded in practice after observing the results derived from it.

The use of co-occurrence matrixes proved to be non-discriminating enough as long as other parts of the image (that do not contain a car) can trigger the attention mechanism since they exhibit similar co-occurrence based values. The fact is that the information provided by co-occurrence matrixes does not uniquely reflect the 2D structure of a car. Instead, the pre-processed sub-region is directly applied to the input of the classifier, as the set of features that is finally used for learning. The dimensions of the entire region of interest are normalized before being fed to the classifier. A size of 70x80 pixels has been chosen as depicted in figure 4. This size is adequate for detecting vehicles at long distances (up to 80 meters).

Several training sets were created for each sub-region in order to store representative samples in different weather and illumination conditions, as suggested in [8]. This technique allows to learn every separate training set using a specialized Support Vector Machine (SVM) [12] that yields excellent results in practice. Otherwise, the use of a global classifier would demand for excessive generalization of the classifier. General classifiers are doom to failure in practice when dealing with images acquired in

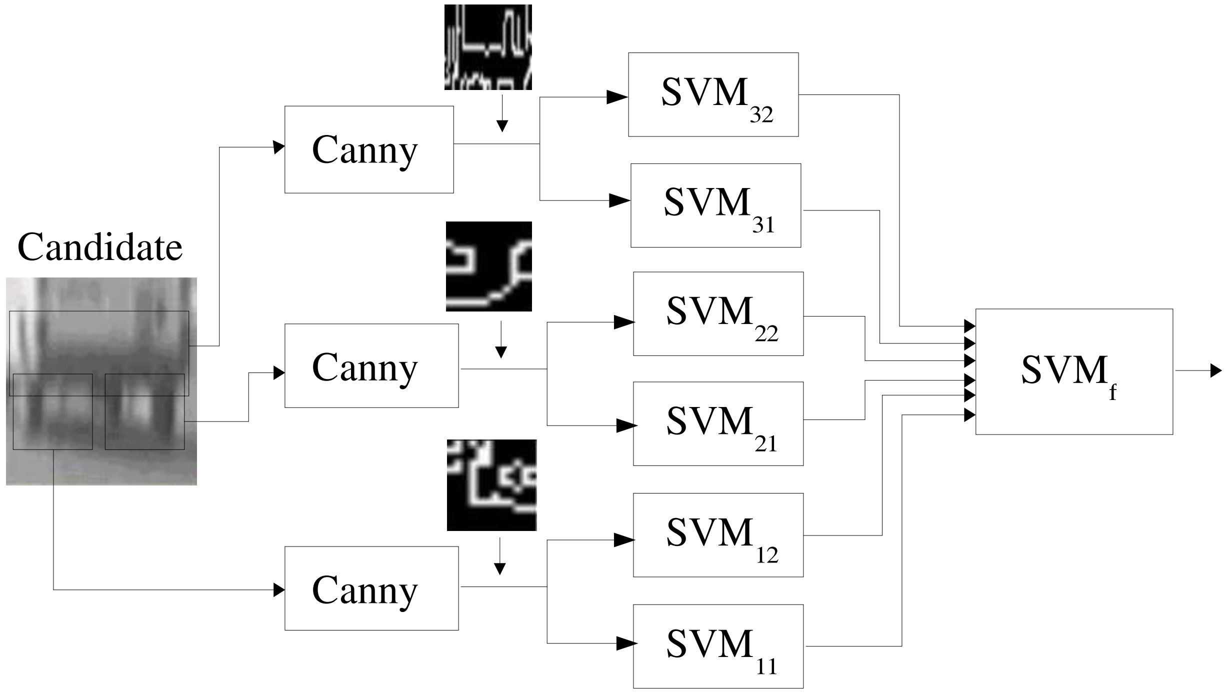

Fig. 5. Global structure of the two-stage SVM classifier.

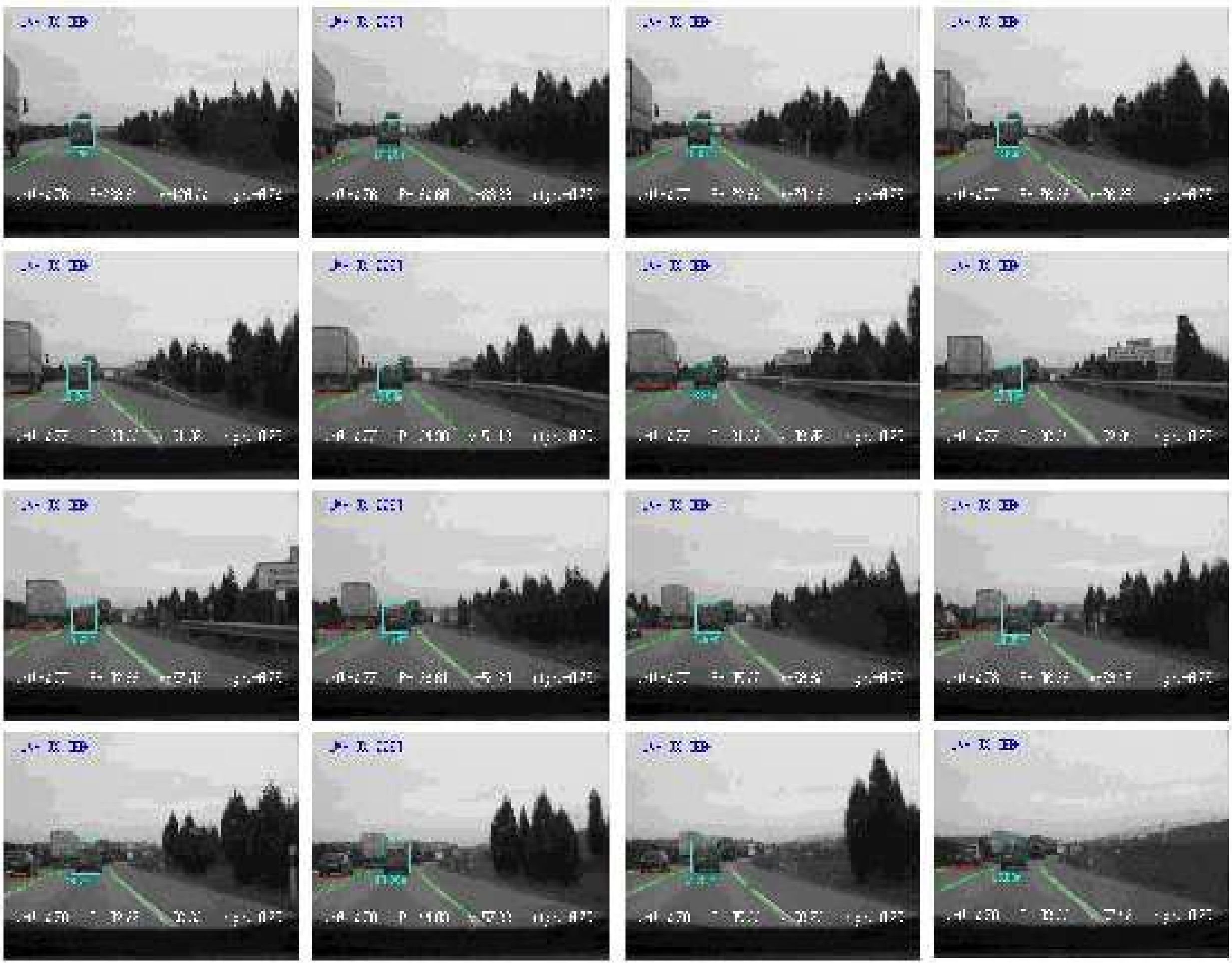

Fig. 6. Vehicle detection and tracking in a sequence of images.

outdoor scenarios, as they contain a huge variability. The global training strategy is carried out in two stages. In a first stage, separate SVM-based classifiers are trained using individual training sets that represent a subset of a sub-region. Each SVM classifier produces an output between -1 (non-vehicle) and +1 (vehicle). Accordingly, it can be stated that this stage provides classification of individual parts of the candidate sub-regions. In a second step, the outputs of all classifiers are merged in a single SVM classifier in order to provide the final classification result. Figure 5 depicts the global structure of the classification process. SVM classifiers in the first stage are denoted as SVM_{ij} in figure 5, where i stands for the sub-region of interest and j represents the training sub-set for that sub-region. The SVM classifier in the second stage is denoted as SVM_f (final SVM).

IV. Results

The system was implemented on a Apple PC at 2.0 GHz running the Knoppix Linux Operating System. The complete algorithm runs at 25 frames/s. We created a database containing 2000 samples of road vehicles. The samples were extracted from recorded images acquired in real experiments onboard a road vehicle in real traffic conditions in Madrid. 2 different training sets were built for the same sub-region in different conditions. This yields a total of 6 training sets (2x3). All training sets were created at day time conditions using the TsetBuilder tool[13], specifically developed in this work for this purpose. By using the TsetBuilder tool different candidate regions are manually selected in the image on a frame-by-frame basis. This allows to select candidate regions containing vehicles of different size, from different manufacturers, and so on. The number of non-vehicle samples in the training sets was chosen to be similar to the number of vehicle samples. Special attention was given to the selection of non-vehicle samples. If we select simple non-vehicle examples (for instance, road regions) the system learns very quickly but it does not develop enough discriminating capability in practice, as the attention mechanism can select a region of the image that might be very similar to a car but it is not a car in reality. The training of all SVM classifiers was performed using the free-licence LibTorch libraries for Linux. We obtained a detection rate of 85% in a test set containing 1000 images, and a false detection rate of 5%. The performance of the single-frame recognition process is largely increased by using multi-frame validation. The probability of a candidate region being classified as vehicle is modelled as a Bayesian random variable. Accordingly, its value is recomputed at each frame as a function of the outputs provided by the single-frame classifier and by a Kalman filter used for vehicle tracking. As an example, figure 6 shows a sequence of images in which a vehicle is detected and tracked along the lane of the host vehicle. A blue box is overprinted over the detected vehicle indicating the estimated distance measured from the host vehicle. Other vehicles appearing along the adjoining lane are marked with a horizontal red line.

V. Conclusions and Future Work

We have developed a visual multi-frame two-stage vehicle classification system based on Support Vector Machines (SVM). The complete system is implemented in C language under Linux Operating System (Knoppix). The learning process has been simplified by decomposing the candidate regions into 3 local sub-regions that are easily learned by individual SVM classifiers. Several training sets have been built for each sub-region in order to cope with different weather and illumination conditions. The complete classifier can be regarded as a hierarchical SVM classifier. The results achieved up to date with a set of 2000 samples are encouraging. Nevertheless they still need to be improved before being safely used as an assistance driving system onboard road vehicles in real conditions. For this purpose, the content of the training sets will be largely increased by including new and more complex samples that will boost the classifier performance, in particular when dealing with difficult cases. We aim at enhancing the classifier ability to discriminate those cases by incorporating thousands of them in the database. In addition, the attention mechanism will be refined in order to provide more candidates around the original candidate region. This will reduce the number of candidate regions that only contain a part of the vehicle, i.e., those cases in which the entire vehicle is not completely visible in the candidate region due to a misdetection of the attention mechanism.

Acknowledgment

This work has been funded by Research Projects CICYT DPI2002-04064-05-04 and FOM2002-002 (Ministerio de Fomento, Spain).

References

[1] G. R. Widman, W. A. Bauson, and S. W. Alland, "Development of collision avoidance systems at Delphi Automotive Systems," *In Proc. Int. Conf. Intelligent Vehicles*, pp. 353-358, 1998.

[2] T. Williamson and C. Thorpe, "Detection of small obstacles at long range using multibaseline stereo," *In Proc. Int. Conf. Intelligent Vehicles*, pp. 311-316, 1998.

[3] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert, "Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner," *In Proc. Int. Conf. On Advanced Robotics*, pp. 1538-1543, 2003.

[4] A. Broggi, M. Bertozzi, A. Fascioli, C. Guarino Lo Bianco, and A. Piazzi, "The Argo autonomous vehicle vision and control systems," *International Journal of Intelligent Control and Systems*, Vol. 3, No. 4, pp. 409-441, 2000.

[5] B. Steux and Y. Abramson, "Robust real-time on-board Vehicle Tracking System using Particles Filter," *In Proc. 5th IFAC Intelligent Autonomous Vehicles*, Lisboa, Portugal, June 2004.

[6] G. P. Stein, O. Mano, and A. Shashua, "Vision-based ACC with a single camera: bounds on range and range rate accuracy". *In Proc. Int. Conf. Intelligent Vehicles*, Versailles, France, June 2002.

[7] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Transactions on Pattern Analisis and Machine Intelligence*, Vol. 23, No. 4, April 2001.

[8] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems: single-frame classifi cation and system level performance," *In Proc. IEEE Intelligent Vehicles Symposium*, pp. 1-6, Parma, Italy, June 14-17, 2004.

[9] G. Grubb, A. Zelinsky, L. Nilsson, and M. Rilbe, "3D Vision sensing for improved pedestrian safety," *In Proc. IEEE Intelligent Vehicles Symposium*, pp. 19-24, Parma, Italy, June 14-17, 2004.

[10] C. Papageorgiou and T. Poggio, "A trainable system for object detection". *Intl J. Computer Vision*, Vol. 38, No. 1, pp. 15-33, 2000.

[11] E. D. Dickmanns and B. D. Mysliwetz, “Recursive 3-D Road and Relative Ego-State Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, February 1992.
[12] J. C. Christopher, “A Tutorial on Support Vector Machines for Pattern Recognition”. *Data Mining and Knowledge Discovery*, No. 2, pp. 121-167. Kluwer Academic Publishers.1. 1998.
[13] J. Nuevo, “TestBuilder Tutorial”. Technical Report 2005. *ftp://www.depeca.uah.es/pub/vision/SVM/manual.pdf*