A Monocular Solution to Vision-Based ACC in Road Vehicles

Miguel Ángel Sotelo, Jesús Nuevo, Manuel Ocaña, and Luis Miguel Bergasa

Department of Electronics, University of Alcalá, Alcalá, Madrid, Spain
{jnuevo, michael, mocana, bergasa}@depeca.uah.es

Abstract. This paper describes a monocular vision-based Adaptive Cruise Control (ACC) System in the framework of Intelligent Transportation Systems (ITS) technologies. The challenge is to use a single camera as input, in order to achieve a low cost final system that meets the requirements needed to undertake serial production.

1 Introduction

A monocular imaging device (a single FireWire digital camera) is deployed to provide "indirect range" measurements using the laws of perspective. Some previous developments 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 color features [4], or pattern recognition techniques [5]. In the current work, the searching space is reduced in an intelligent manner in order 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 find 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 and vehicle detection systems is provided below, together with some graphical results.

2 System Description

2.1 Lane Tracking

The system is divided in three modular subsystems with specific functions. The first subsystem is responsible for lane detection and tracking, as well as lane crossing monitoring. Images obtained from the camera are processed and clothoid curves are fitted to the detected markings. The algorithm scans up to 25 lines in the *area of interest*, 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 unstabilities in the fitted curve. The final state vector is composed of 6 variables [7] for each line on the road: c_{oh}, c_{1h}, c_{ov}, c_{1v}, x_o, ψ_o, where c_{oh} and c_{1h} represent the clothoid

R. Moreno Díaz et al. (Eds.): EUROCAST 2005, LNCS 3643, pp. 507 – 512, 2005.

horizontal curvature parameters, c_{ov} and c_{1v} stand for the clothoid vertical curvature parameters, while x_o and ψ_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 *area of interest*. Figure 1 depicts a sequence of images in which the result of the lane tracking algorithm is overprinted on the road images.

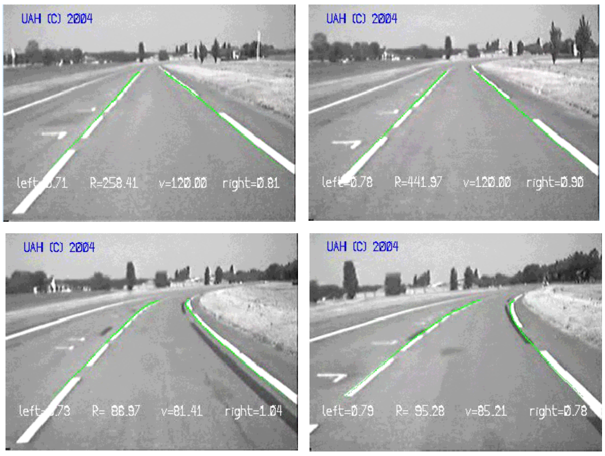

Fig. 1. Lane tracking example in a sequence of images. The green lines represent the estimated lines of the road. The example also depicts the error between the left wheel of the car the the left lane (left), the error between the right wheel of the car and the right lane (right), the radious of curvature of the road estimated at a lookahead distance of 50m (R), and the maximum recommended velocity to bend the curve (V) according to the radious of curvature.

2.2 Car Detection and Recognition

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, as depicted in figure 2, 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. 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.

Fig. 2. Sequential vehicle candidates searching along the detected lane

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

A set of features must be extracted from each sub-region and fed to the classifier. Before doing that, the 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. 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) [6] 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 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.

3 Results and Conclusions

The system was implemented on a Power Mac 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. All training sets were created at day time conditions using the TsetBuilder tool [9], 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. 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 based on a Kalman filter. As an example, figure 3 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. The distance between the ego-vehicle and the preceding vehicle along the lane becomes the input to the Adaptive Cruise Control (ACC) System.

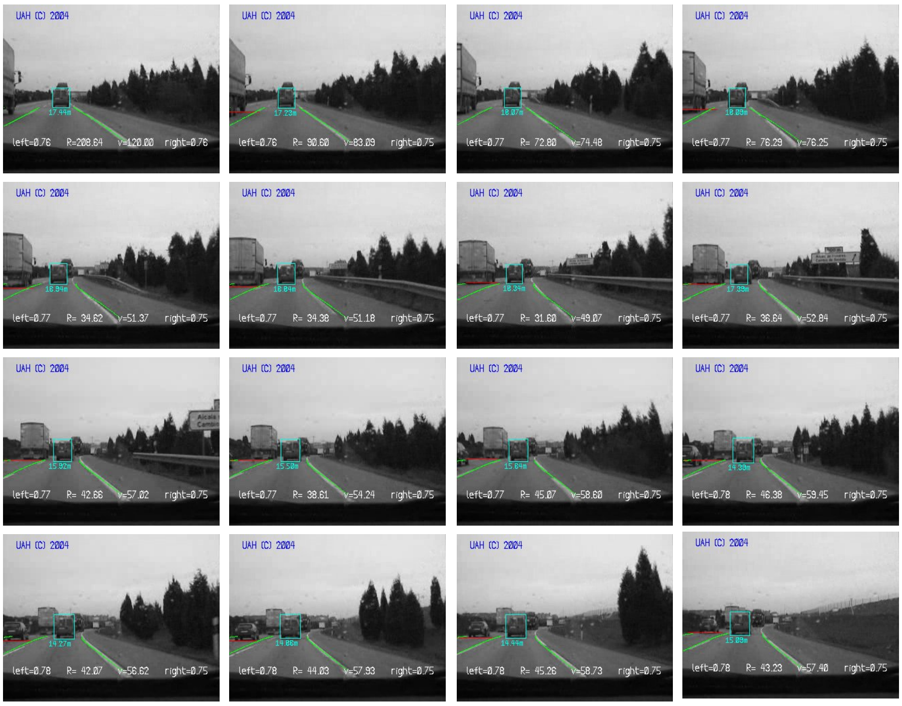

Fig. 3. Vehicle tracking example in a sequence of images

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.

Acknowledgments

This work has been funded by Research Projects CICYT DPI2002-04064-05-04 (*Ministerio de Educación y Ciencia*) 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's vision and control systems". International Journal of Intelligent Control and Systems. Vol. 3, No. 4, 409-441, 2000.
5. 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, 2002.
6. J. C. Christopher Burges, "A Tutorial on Support Vector Machines for Pattern Recognition". Data Mining and Knowledge Discovery, 2,121-167 (1998). Kluwer Academic Publishers.1.
7. 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.
8. A. Shashua, Y. Gdalyahu, and G. Hayun. "Pedestrian detection for driving assistance systems: single frame classification and system level performance". In Proc. IEEE Intelligent Vehicles Symposium, pp:1-6, Parma, Italy, June 14-17 2004.
9. J. Nuevo. "TestBuilder Tutorial". Technical Report, Department of Electronics, University of Alcalá, 2005. ftp://www.depeca.uah.es/pub/vision/SVM/manual