A Cooperative Approach to Vision-based Vehicle Detection

A. Bensrhair, M. Bertozzi, A. Broggi, P. Miché, S. Mousset, and G. Toulminet

Abstract— **In this paper two different vision based systems for vehicle detection are described and their integration discussed. The first approach is based on the use of a specific model for vehicles and mostly relies on monocular vision. Conversely, the second system is based on the use of stereo vision and allows to refine the coarse results obtained by the former.**

A preliminary integration of the two systems has been tested on the ARGO experimental vehicle and some remarks about reliability and robustness are also included.

Keywords— **stereo vision, data fusion, vehicle detection**

I. Introduction

Several vision-based approaches are used for the detection of obstacles in an automotive environment [1]. Depending on the definition of obstacle the techniques exploited for the detection may vary. In case only vehicles are to be detected, specific patterns can be used for the search, for example: shape [2], symmetry [3], texture [4], or the use of an approximant contour [5]. In such a case the processing can be reduced to the analysis of a single still image. While this approach has been widely demonstrated to be effective for a mere vehicle detection, it is difficult to accurately determine the vehicle distance. Moreover, in the case of single image processing, specific patterns on the scene (e.g. shadows, lane markings, or other artifacts on the road surface) can potentially confuse the vision system.

A more challenging task is the detection of *any* object that can obstruct the vehicle's driving path, namely a generic obstacle. In such a case, more complex techniques are used, mostly being based on the processing of two or more images, such as the optical flow field analysis [6, 7] or the use of stereovision [8, 9]. These techniques feature a higher computational complexity mainly due to the higher amount of data to be processed. In addition, they must also be robust enough to tolerate noise caused by vehicle movements and drifts impacting on the calibration of the vision system.

This work presents the integration of the vision-based systems for vehicle detection developed by the Universities of Parma and Rouen. The former is based on the processing of monocular images and the use of a specific model for vehicles. The results of the computation are fed to the latter that, conversely, is based on the use of stereo-vision and does not rely on a specific model for obstacles. Both systems have been installed and tested on ARGO, an experimental vehicle equipped for testing vision algorithms and autonomous driving [5].

This work has been supported by the Galileo Program

A. Bensrhair, P. Miché, S. Mousset, and G. Toulminet are with the Université de Rouen et INSA de Rouen, France. E-mail: {abdelaziz.bensrhair,pierre.miche,stephane.mousset,gwenaelle.toulminet}@insa-rouen.fr.

M. Bertozzi is with the Dip. di Ingegneria dell'Informazione, Università di Parma, Italy. E-mail: bertozzi@ce.unipr.it.

A. Broggi is with the Dip. di Informatica e Sistemistica, Università di Pavia, Italy. E-mail: alberto.broggi@unipv.it.

This paper is organized as follows: section II briefly depicts the two different stereo vision systems; section III details the algorithms used; results and timings performance are discussed in section IV, while section V ends the paper with some final remarks.

II. Vision Systems Details

A. University of Parma vision system

Two small (3.2 $cm \times 3.2$ cm) cameras are used to synchronously acquire pairs of grey level images. They feature a 6.0 mm focal length and a 360 lines resolution and receive the synchronism from an external signal generator.

The cameras are installed inside ARGO behind the top corners of the windscreen (see figure 1), thus maximizing the longitudinal distance between the two cameras. The camera optical axes are parallel and, in order to handle the detection of tall vehicles, part of the scene over the horizon is captured, even if the framing of a portion of the sky can be critical for image brightness: in case of high contrast the sensor may happen to acquire oversaturated images.

Fig. 1
The University of Parma vision system installed into ARGO.

The images are acquired by a PCI Matrox board, which is able to grab three 768×576 pixel images simultaneously. They are directly stored into the main memory of the host computer thanks to the use of DMA and PCI bus-mastering. The computing engine used for this experiment is a Pentium II 450 MHz PC with Linux OS. The acquisition can be performed in real time, at a 25 Hz rate in case of full frames or at a 50 Hz rate in case of single field acquisition.

B. University of Rouen stereovision system

The University of Rouen has designed a passive stereovision sensor made up of a rigid body, two similar lenses and two Philips VMC3405 camera modules whose centers are separated

0-7803-7194-1/01/$10.00 2001 IEEE

by 12.7 cm (figure 2). During the experiment, two different set of lenses have been used: 16 mm and 50 mm to test its behavior in correspondence to different vehicle distances.

Fig. 2
THE UNIVERSITY OF ROUEN SENSOR INSTALLED INTO ARGO.

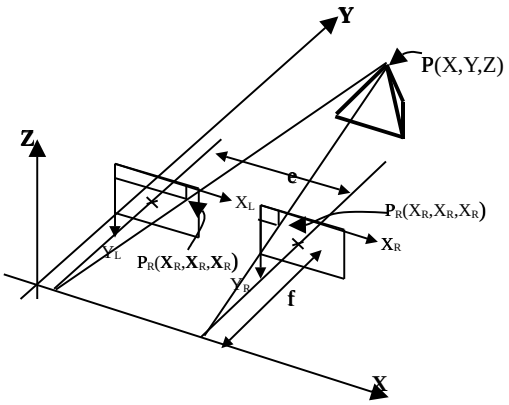


Fig. 3
THE CONFIGURATION OF THE STEREOVISION SENSOR.

An Imaging Technology PC-RGB frame grabber controls these two cameras, and acquires simultaneously their two images (720×568 or 720×284 pixels). The clock on the frame grabber AD-converter is the pixel clock of one of the two cameras. It is a timing signal which is used to divide the incoming lines of the video signals into pixels. With such a clock maximum resolution can be reached and alias effects are avoided. Furthermore, the two camera-lens units are set up so that their optical axes are parallel and, in order to respect the epipolar constraint, the straight line joining the two optical centers is parallel to each images horizontal line.

Based on the epipolar configuration of this sensor, depth information is given in meters by:

$$Z = \frac{f \times e}{p \times \delta} \tag{1}$$

where e is the distance between the two optical centers, p is the width of the CCD pixel, f is the focal length of the two lenses. δ is given in pixels and is the horizontal disparity of two stereo-corresponding points. Let P_L and P_R be two stereo-corresponding points of a 3D point P of an object (figure 3). Let (X_L, Y_L), (X_R, Y_R) and (X, Y, Z) be their coordinates. (X_L, Y_L) and (X_R, Y_R) are given in pixels, (X, Y, Z) is given in meters. Then, due to the epipolar configuration, $Y_L = Y_R$ and $\delta = (X_R - X_L)$.

The architecture used for this experiment is a Pentium III 800 MHz with the Windows OS.

III. ALGORITHMS FOR VEHICLE DETECTION

In this section the monocular and stereo approaches to vehicle detection are detailed.

A. Monocular phase

A vehicle, generally, features a high degree of symmetry (when framed from the rear) and is characterized by a rectangular bounding box with a specific aspect-ratio. Initially, an area of interest is identified on the basis of perspective constraints and searched for possible vertical symmetries. Once the symmetry position and width have been detected, a new search begins, aimed at the detection of the two bottom corners of a simplified vehicle model, namely a rectangular bounding box. Since this approach relies on monocular vision only, an approximated distance is computed.

A.1 Symmetry detection

In order to determine the symmetry content of acquired images a *symmetry map* is used. The symmetry map is an image whose pixels encode the symmetry content. The horizontal coordinate of each pixel refers to the position of the vertical symmetry axis within the area of interest. The vertical coordinate is related to the horizontal width of the image area considered for computing the symmetry. The brighter the pixel the higher the symmetry.

The analysis of gray level images only does not suffice for determining all symmetrical features. In order to increase the detection robustness, also vertical and horizontal edges are extracted, thresholded, and symmetries are computed into these domains as well.

A combined symmetry map is computed as a weighted sum of the symmetry map obtained from the grey level image and the ones obtained by the analysis of horizontal and vertical edges. Figure 4 shows both the partial symmetry maps computed starting from grey level and edges images and their weighted combination.

A.2 Model matching

The symmetry map identifies a specific region of interest in which a model of a vehicle, a rectangular bounding box, is looked for. This model is detected through a search for its corners. Initially, the symmetrical region in the edge image is checked for the presence of two corners representing the bottom of the bounding box. The presence of corners is validated using perspective and size constraints [10].

This process is followed by the detection of the top part of the bounding box, which is looked for in a specific region whose location is again determined by perspective and size constraints.

A backtracking approach is used in case no valid bounding boxes are found in correspondence to the symmetry maximum. This situation is generally due to the presence of background symmetrical patterns. The following local maxima are considered and the search for a bounding box is performed again.

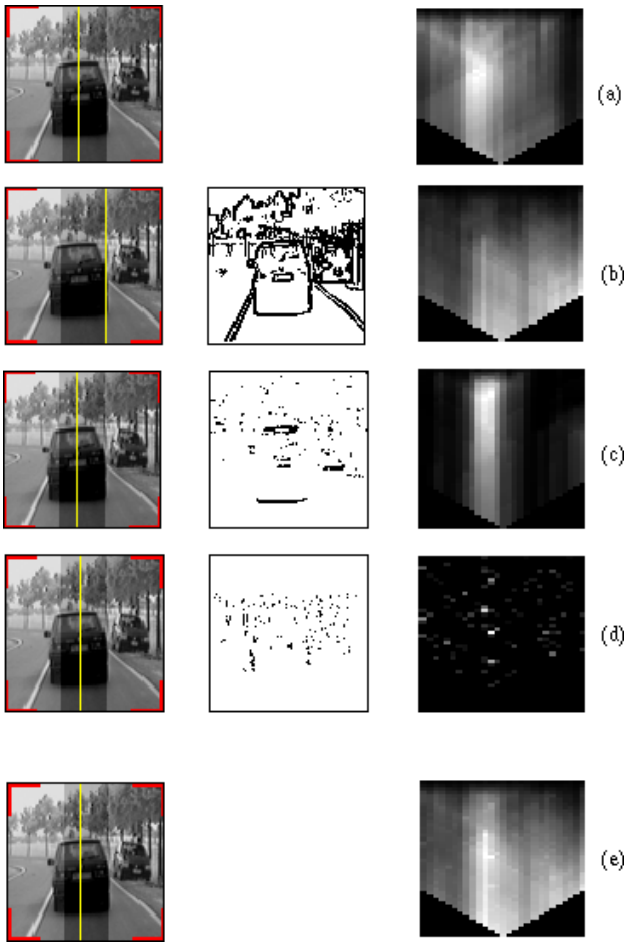


Fig. 4

COMPUTING THE RESULTING SYMMETRY: (*a*) GREY-LEVEL SYMMETRY; (*b*) EDGE SYMMETRY; (*c*) HORIZONTAL EDGES SYMMETRY; (*d*) VERTICAL EDGES SYMMETRY; (*e*) TOTAL SYMMETRY. FOR EACH ROW THE RESULTING SYMMETRY AXIS IS SUPERIMPOSED ONTO THE LEFTMOST ORIGINAL IMAGE.

A.3 Distance computation

Thanks to the knowledge of the vision system calibration it is possible to compute the distance from the leading vehicle. Figure 5 shows the output of this monocular phase and the distance computed by the system relying on monocular vision only.

Unfortunately, it may happen that the lower part of the vehicle is not correctly detected, therefore leading to wrong values for vehicle distance. Sometimes, in fact, the luminance gradient of the region between the rear bumper and the chassis is so high to be misinterpreted as the lower part of the vehicle. In order to refine this measurement an adjustment step is mandatory.

A stereo vision approach to distance refinement has already been developed and described in [10]. Anyway, in the following paragraph we present an alternate solution based on a second vision system able not only to refine the result but to validate it too.

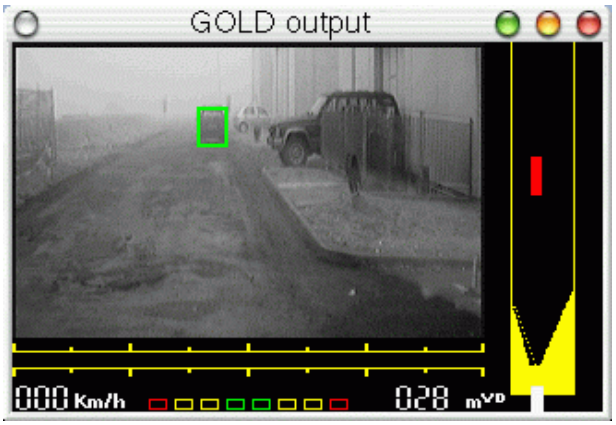


Fig. 5

VEHICLE DETECTION RESULTS OF THE MONOCULAR PHASE: A BRIGHT BOUNDING BOX IS SUPERIMPOSED ON THE ACQUIRED IMAGE WHERE THE VEHICLE IS DETECTED AND ALSO THE COMPUTED DISTANCE (28 *m*, IN THIS CASE) IS SHOWN. ON THE RIGHT, A RECONSTRUCTION OF THE ROAD SEEN FROM THE TOP DEPICTS THE POSITION OF THE VEHICLE WITHIN THE AREA IN FRONT OF THE VISION SYSTEM.

B. The stereovision phase

The result of the detection discussed in the previous phase is used to construct a depth map of the vehicle. As a consequence, we have to find the projection of the rectangular bounding box that characterizes the vehicle detected by ARGO in our depth map. To do this, the cooperation process is made up of two major parts. The result of the previous processing is cropped along the vehicle bounding box. The data used for the cropping are: vehicle distance D, width W, height H, and the coordinates (X_P, Y_P, D) of bounding box bottom midpoint P. In the second part, using the 3D points previously computed, 3D curves are built. Some criteria are used to select 3D curves belonging to the vehicle. Then, the smallest rectangle that contain the remaining 3D curves is considered to be the projection of the vehicle bounding box.

The complete process is presented in figure 6 and described in the following section.

B.1 Segmentation and cropping

The segmentation step uses a self-adaptive and monodimensional operator, the *declivity* [11]. Declivity is defined as a set of consecutive pixels in an image line, whose grey levels are a strictly monotonous function of their positions. Each declivity is characterized by its amplitude defined by: $d_i = I(x_{i+1}) - I(x_i)$.

Relevant declivities are extracted by thresholding these amplitudes. To be self-adaptive, the threshold value is defined by [11]:

$$d_t = 5.6 \times \sigma \qquad (2)$$

where σ is the standard deviation of the component of a white noise which is supposed Gaussian and calculated by using the histogram of grey levels variations of pixels in an image line. The coefficient value is fixed in order to reject 99.5% of increments due to noise. In order to have a good depth map accuracy,

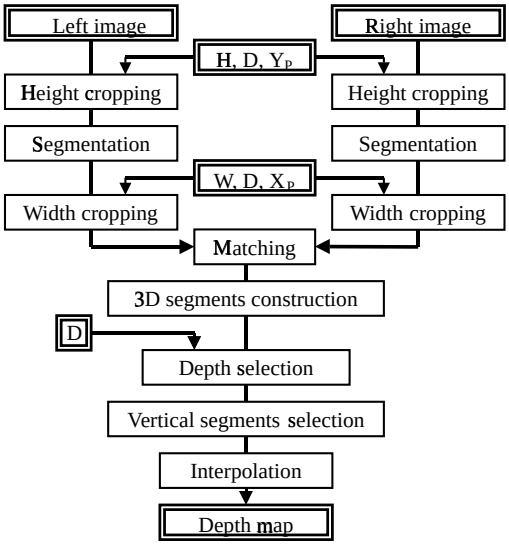


Fig. 6

STEREOVISION PHASE PROCESS: SINGLE FRAMED BOXES INDICATE A STEP IN THE PROCESS, WHILE DOUBLE FRAMED BOXES SHOW THE INPUT DATA AND THE FINAL OUTPUT.

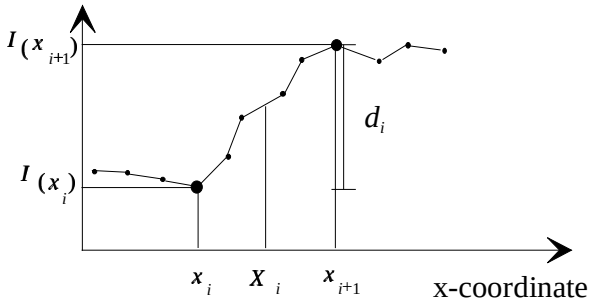


Fig. 7

CHARACTERISTICS PARAMETERS OF A DECLIVITY.

efficient locations of relevant declivities are essential. The position of a declivity is calculated using the mean position of the declivity points weighted by the gradients squared:

$$X_i = \frac{\sum_{x=x_i}^{x_{i+1}-1} [I(x+1)-I(x)]^2 (x+0.5)}{\sum_{x=x_i}^{x_{i+1}-1} [I(x+1)-I(x)]^2} \quad (3)$$

where X_i is the position of the declivity on an image line as shown in figure 7.

Before the segmentation step, right and left grey level images are only cropped in height, in order to have a good estimation of σ ; After the segmentation step and before the matching process, right and left declivity maps are cropped in width (figure 6). The height of the two final frames is $h+2\delta_h$ and their widths are $w+2\delta_w$, where h and w are the height and width, in pixels, of the vehicle bounding box. δ_H and δ_W are inserted to compensate data inaccuracy. Let be (x_{rP}, y_P) and (x_{lP}, y_P) the coordinates of the projection of point P in the right and left images. Thus, due to the sensor configuration, left and right grey level images are segmented line by line, from line $y_P - \delta_H$ to line $y_P + h + \delta_H$. Concerning width cropping, we just keep left relevant declivities from column $x_{lP} - \frac{w}{2} - \delta_W$ to column $x_{lP} + \frac{w}{2} + \delta_W$ in the left declivity map; and right relevant declivities from column $x_{rP} - \frac{w}{2} - \delta_W$ to column $x_{rP} + \frac{w}{2} + \delta_W$ in the right declivity map.

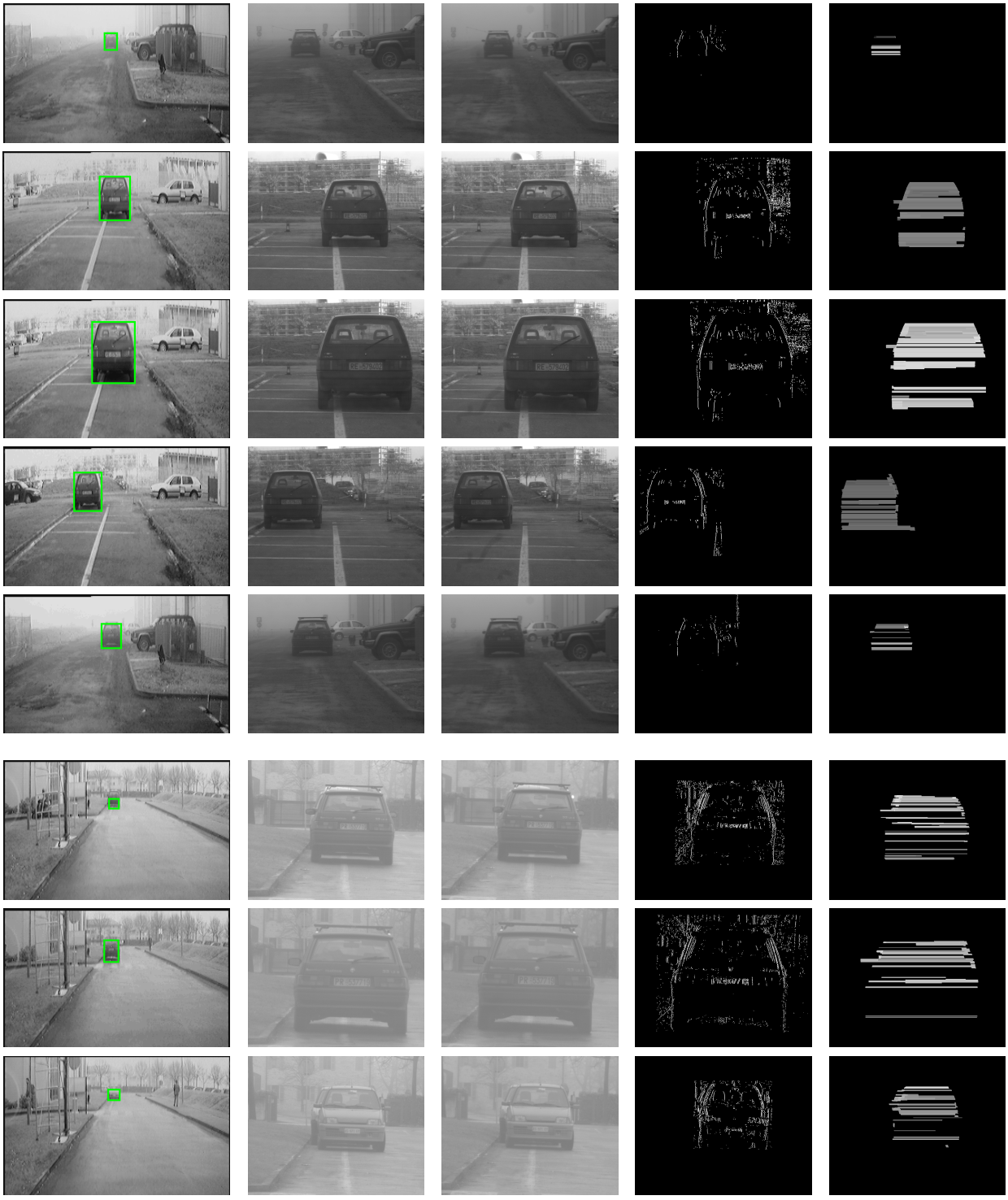
B.2 The matching algorithm

The matching algorithm provides depth information, based on the positions of left and right relevant declivities, by using a dynamic programming method. Due to the configuration of the stereo vision sensor, it is done line by line. Then, the matching problem can be summarized as finding an optimal path on a two-dimensional graph whose vertical and horizontal axes respectively represent the declivities of a left line and the declivities of the stereo-corresponding right line. Axes intersections are nodes that represent hypothetical declivity associations. Optimal matches are obtained by the selection of the path which corresponds to a maximum value of a global gain. It is computed by using local gains which represent the qualities of hypothetical declivity associations. Local gain function is non-linear. Thus, the matching algorithm is self-adaptive, robust and fast.

The matching algorithm consists of three steps. In the first step, we construct all possible declivity associations taking into consideration geometric and photometric constraints. For each declivity association, we calculate a local gain. In the second step, nodes corresponding to hypothetical associations are positioned on the graph. During graph construction, several paths are also constructed from initial nodes to intermediate nodes. In the last step, among all the final nodes generated, we chose the one that corresponds to a maximum global gain. Then, starting from this node and until the initial nodes is reached, we move up the graph following the optimal path and applying order and uniqueness constraints. The nodes obtained are correct declivity associations whose disparity is calculated. The result of the matching algorithm is a 3D edge points map.

B.3 Processing 3D curves

By means of line by line processing, 3D curves are made based on 3D points, using relatedness and depth criteria. Then, small curves and curves that have no points whose depth is between $D+\Delta_D$ and $D-\Delta_D$ are eliminated, where Δ_D is inserted to compensate depth inaccuracy. As road environment is structured, 3D curves can be approximated by means of one or several 3D straight segments. Since our purpose is vehicle detection, we only retain the 3D segments which are almost vertical in the image. So, by an iterative partition method, 3D curves are decomposed into 3D segments whose slopes in the image are calculated by a least square method. Then, 3D curves whose 3D segments are not almost vertical are eliminated. Finally, the image is cropped. The new frame is the smallest rectangle that contains all the 3D curves. Because the vehicle has two sides, from this frame we only keep lines starting from the first one that contains at least two 3D points to the last one that also contains two 3D points. In order to achieve depth map of the vehicle detected during the monocular phase, an interpolation step is used. At the end, for each 3D curves, the depth mean value of its 3D points is calculated. The closest curve is the depth of the vehicle calculated by the stereovision system.



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

Fig. 8

RESULTS OF VEHICLE DETECTION: (*a*) IMAGE ACQUIRED BY THE MONOCULAR VISION SYSTEM WITH THE SUPERIMPRESSION OF A BOUNDING BOX DETECTING THE VEHICLE, (*b*) AND (*c*) IMAGES ACQUIRED BY THE STEREO VISION SYSTEM, (*d*) DECLIVITY, AND (*e*) DEPTH MAP.

IV. RESULTS

In order to evaluate the performance of the two phases and determine possible enhancements, an extensive test has been carried out. A target vehicle has been positioned in front of the ARGO at given distances and the measure of the distance computed.

Figure 8 shows few images and results of the test: column (a) displays the result of the computation of the monocular phase,

namely a bounding box superimposed on the original image encoding both distance and size of the detected obstacle, (b) and (c) present the two images acquired by the stereo vision system using two different lenses sets (16 mm focal for the five upper rows and 50 mm for the other rows), while (d) shows the declivity computed on the portion of stereo images (b) and (c) that contains the detected vehicle; the final result, namely a depth map of the vehicle, is shown in (e).

Table I presents a number of results showing the computed distances and comparing them with the actual distance.

TABLE I

DISTANCES COMPUTED BY THE TWO PHASES.

	Distances (m)							
Mono Phase	9.7	13.5	15.6	21.7	20.7	27.1	34.6	28.3
Stereo Phase	10.3	16.8	18.8	21.4	31.9	26.6	36.5	53.2
Actual	14.4	16.4	18.4	25.0	30.0	34.4	40.3	45.0

The monocular vision phase takes nearly 20 ms on a 450 MHz Pentium II architecture. The stereo vision phase requires 300 ms on a 800 MHz Pentium III machine processing the entire image; conversely, it needs around 250 ms in case only the portion of the image that contains the vehicle is analyzed; it is to be noticed that the code used for the stereo vision phase has not yet been optimized.

V. DISCUSSION

In this work a cooperative system for vehicle detection has been presented. It is based on two separate phases: a first one relying on monocular vision only followed by a second processing based on stereo vision.

The main target of this work is to exploit the best of each approach and to overcome the weak points of each phase. In fact monocular vision is not as effective as stereo vision in recovering vehicles distance, but, at the same time, stereo vision requires to process a larger amount of data thus being implicitly slower.

The monocular phase allows to select a reduced portion of the image where a vehicle is detected. The subsequent stereo vision processing is performed on a reduced portion of the scene, therefore speeding up the process.

The next research steps include a more strict integration (hardware and software) between the two systems. The stereo vision phase, thanks to its superior precision, can validate and refine the result of the monocular phase, allowing the detection of mistakes in the first processing or the development of a tracking phase.

REFERENCES

[1] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based Intelligent Vehicles: state of the art and perspectives," *Journal of Robotics and Autonomous Systems*, vol. 32, pp. 1–16, June 2000.

[2] F. Thomanek, E. D. Dickmanns, and D. Dickmanns, "Multiple Object Recognition and Scene Interpretation for Autonomous Road Vehicle Guidance," in *Procs. IEEE Intelligent Vehicles Symposium'94*, (Paris), pp. 231–236, Oct. 1994.

[3] A. Kuehnle, "Symmetry-based vehicle location for AHS," in *Procs. SPIE - Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*, vol. 2902, (Orlando, FL), pp. 19–27, Nov. 1998.

[4] T. Kalinke, C. Tzomakas, and W. von Seelen, "A Texture-based Object Detection and an Adaptive Model-based Classification," in *Procs. IEEE Intelligent Vehicles Symposium'98*, (Stuttgart, Germany), pp. 341–346, Oct. 1998.

[5] A. Broggi, M. Bertozzi, A. Fascioli, and G. Conte, *Automatic Vehicle Guidance: the Experience of the ARGO Vehicle*. World Scientific, Apr. 1999. ISBN 981-02-3720-0.

[6] T. Suzuki and T. Kanade, "Measurement of Vehicle Motion and Orientation using Optical Flow," in *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems'99*, (Tokyo, Japan), pp. 25–30, Oct. 1999.

[7] W. Kruger, W. Enkelmann, and S. Rossle, "Real-Time Estimation and Tracking of Optical Flow Vectors for Obstacle Detection," in *Procs. IEEE Intelligent Vehicles Symposium'95*, (Detroit, USA), pp. 304–309, Sept. 1995.

[8] A. Bensrhair, P. Miché, and R. Debrie, "Fast and automatic stereo vision matching algorithm based on dynamic programming method," *Pattern Recognition Letters*, vol. 17, pp. 457–466, 1996.

[9] U. Franke, "Real-Time Stereo Vision for Urban Traffic Scene Understanding," in *Procs. IEEE Intelligent Vehicles Symposium 2000*, (Detroit, USA), pp. 273–278, Oct. 2000.

[10] A. Broggi, M. Bertozzi, A. Fascioli, C. Guarino Lo Bianco, and A. Piazzi, "Visual Perception of Obstacles and Vehicles for Platooning," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, pp. 164–176, Sept. 2000.

[11] P. Miché and R.Debrie, "Fast and self-adaptative image segmentation using extended declivity," *Annals of telecommunication*, vol. 50, no. 3-4, pp. 401–410, 1995.