

# Visual Perception of Obstacles and Vehicles for Platooning

Alberto Broggi, *Member, IEEE*, Massimo Bertozzi, *Member, IEEE*, Alessandra Fascioli, *Member, IEEE*, Corrado Guarino Lo Bianco, and Aurelio Piazzi, *Member, IEEE*

**Abstract**—This paper presents the methods for sensing obstacles and vehicles implemented on the University of Parma experimental vehicle (ARGO). The ARGO project is briefly described along with its main objectives; the prototype vehicle and its functionalities are presented. The perception of the environment is performed through the processing of images acquired from the vehicle. Details about the stereo vision-based detection of generic obstacles are given, along with a measurement of the performance of the method; then a new approach for leading vehicles detection is described, relying on symmetry detection in monocular images. This paper is concluded with a description of the current implementation of the control system, based on a gain scheduled controller, which allows the vehicle to follow the road or other vehicles.

**Index Terms**—Automatic steering, image processing, obstacle detection, platooning, vehicle detection, vision-based autonomous vehicles, visual servoing.

## I. INTRODUCTION

AMONG the many functionalities an intelligent vehicle must perform, *Obstacle* and *Vehicle Detection* play a basic role. In fact, an autonomous vehicle must be able to detect vehicles and potential obstacles on its path in order to perform *Road Following*, namely, the automatic movement along a predefined path, or *Platooning*, namely, the automatic following of a preceding vehicle.

A number of different vision-based techniques have been proposed in the literature and tested on prototype vehicles. Several approaches to obstacle detection rely on the localization of specific patterns (possibly supported by features such as shape, symmetry, or edges) and are therefore based on the analysis of monocular images [1], [2]. They generally offer simple algorithmic solutions, allow fast processing and do not suffer from vehicle movements. For example, the research group of the Istituto Elettrotecnico Nazionale “G. Ferraris” limits the processing to the image portion that is assumed to represent the road; borders that could represent a potential vehicle are looked for and examined [3]. People at the Universität der Bundeswehr enforce an edge detection process with obstacle modelization; the system is able to detect and track up to twelve objects around the vehicle [4].

Manuscript received March 7, 2000; revised August 14, 2000. This work was supported in part by the Italian National Research Council (CNR) in the framework of the MADESS2 Project. The Associate Editor for this paper was Dr. Charles E. Thorpe.

A. Broggi is with the Dip. di Informatica e Sistemistica, Università di Pavia, 27100 Pavia, Italy (e-mail: alberto.broggi@unipv.it).

M. Bertozzi, A. Fascioli, C. Guarino Lo Bianco, and A. Piazzi are with the Dip. di Ingegneria dell’Informazione, Università di Parma, 43100 Parma, Italy (e-mail: {bertozzi; fascioli; guarino; aurelio}@ce.unipr.it).

Publisher Item Identifier S 1524-9050(00)10334-5.

Following a more general definition, an obstacle is defined as an object that can obstruct the vehicle’s driving path or, in other words, anything rising out significantly from the road surface. In this case the problem of Obstacle Detection is dealt with using more complex techniques, such as the analysis of the *optical flow* field [5], [6] or the processing of *stereo* images [7]–[9]. As an example, the ASSET-2 [10], [11] is based on optical flow only. Its main feature is that no restrictive assumptions are made about the world, the motion or the calibration of the camera, or other parameters. A different approach has been used for the UTA demonstrator car; in this case a feature-based stereo vision system has been developed and is able to run in real-time even on a 200 MHz powerPC [7].

Besides their intrinsic higher computational complexity, caused by a significant increment in the amount of data to be processed, these techniques must also be robust enough to tolerate the noise caused by vehicle movements and drifts in the calibration of the multiple cameras’ setup. Optical flow-based techniques permit the computation of ego-motion and obstacle’s relative speed, but, as the presence of obstacles is indirectly derived from the analysis of the velocity field, they fail when both the vehicle and obstacle have small or null speed. Conversely, from the analysis of stereo images, obstacles can be directly detected and a three-dimensional (3-D) reconstruction of the environment can be performed. Moreover, to decrease the complexity of stereo vision, some domain specific constraints can be adopted.

On the ARGO autonomous vehicle, obstacle Detection is reduced to the identification of the *free-space* (the area in which the vehicle can safely move). A geometrical transform is performed to detect the free space, using a pair of stereo images of the portion of the road in front of ARGO. This functionality has been thoroughly tested on different obstacles—with varying shape and size—displaced in front of the vehicle in different positions. Results have been collected and analyzed highlighting the characteristics and deficiencies of this approach.

While Obstacle Detection has been proven to be robust allowing ARGO to reliably detect generic obstacles close to the vehicle, the tests demonstrated that its sensitivity is too low in the region far ahead of the vehicle. Therefore, a different approach is needed for finding and following a preceding vehicle. For this reason, a Vehicle Detection functionality has been developed. This functionality is aimed at detecting vehicles only, therefore it is based on a search for specific patterns using shape, symmetry, and size constraints. Vehicles are localized and tracked using a single monocular image sequence whilst a distance refinement is computed using stereo vision. The

steering control for the Platooning functionality is based on a gain scheduled proportional controller whose error input is evaluated using the estimated position of the preceding vehicle.

This paper is organized as follows: the next section presents the ARGO project and the prototype vehicle developed within this framework. Section III presents the Obstacle Detection functionality used in the last few years on ARGO as well as a critical analysis of this algorithm. The Vehicle Detection functionality is addressed in Section IV; Section V presents timings issues, while Section VI describes the control subsystem that drives ARGO. Section VII ends the paper with some remarks.

## II. THE ARGO PROJECT

The main target of the ARGO Project [12] is the development of an active safety system which can also act as an automatic pilot for a standard road vehicle.

The capability of perceiving the environment is essential for an intelligent vehicle which is expected to use the existing road network with no need for specific infrastructures. Although very efficient in some fields of application, active sensors—besides polluting the environment—feature some specific problems in automotive applications due to inter-vehicle interference amongst the same type of sensors, and due to the wide variation in reflection ratios caused by many different reasons, such as obstacles' shape or material. Moreover, the maximum signal level must comply with safety rules, i.e., it must be lower than a safety threshold. For this reason, in the implementation of the ARGO vehicle, the use of sensors has been restricted to passive ones, such as cameras.

A second design choice was to keep the system costs low. These costs include both production costs (which must be minimized to allow a widespread use of these devices) and operating costs, which must not exceed a certain threshold in order not to alter vehicle performance. Therefore, low-cost devices have been preferred, both for image acquisition and processing: the prototype is based on cheap cameras and a commercial PC.

### A. The ARGO Vehicle Prototype

ARGO, shown in Fig. 1, is an experimental autonomous vehicle equipped with vision systems and automatic steering capability.

It is able to determine its position with respect to the lane, to compute road geometry, to detect generic obstacles on the path, and to localize a leading vehicle. The images acquired by a stereo rig placed behind the windshield are analyzed in real-time by the computing system located in the boot. The results of the processing are used to drive an actuator mounted onto the steering wheel and other driving assistance devices.

The system was initially conceived as a safety enhancement unit: in particular it is able to supervise the driver behavior and issue both optic and acoustic warnings or even take control of the vehicle when dangerous situations are detected. Further developments have extended the system functionalities to a complete automatic steering.

1) *The Sensing System:* Only passive sensors are used on ARGO to sense the surrounding environment:

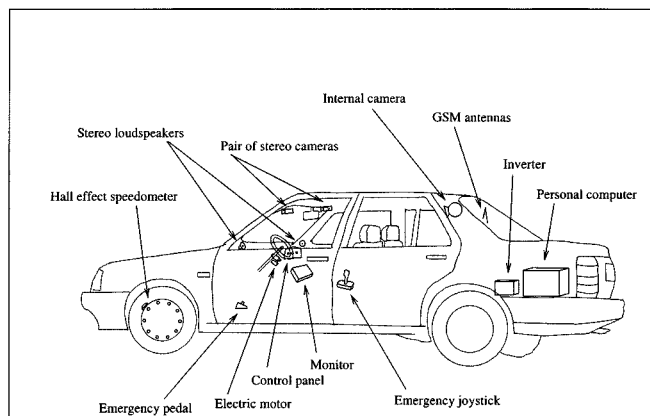


Fig. 1. ARGO prototype vehicle.

- A stereoscopic vision system consisting of two low-cost synchronized cameras able to acquire pairs of grey level images simultaneously. The cameras lie inside the vehicle at the top corners of the windshield, so that the longitudinal distance between the two cameras is maximum.
- A speedometer which is used to detect the vehicle's velocity. A Hall effect device has been chosen for its simplicity and reliability and has been interfaced to the computing system via a digital I/O board.

In addition, a button-based control panel has been installed enabling the driver to modify a few driving parameters, select the system functionality, issue commands, and interact with the system.

2) *The Processing System:* The architectural solution currently installed on the ARGO vehicle is based on a standard 450 MHz Pentium II processor. Thanks to recent advances in computer technologies, commercial systems offer nowadays sufficient computational power for this application. All the processing needed for the driving task (image feature extraction and vehicle trajectory control) is performed in real-time: 50 pairs of single field images are processed every second.

3) *The Output System:* Several output devices have been installed on ARGO: acoustical (stereo loudspeakers) and optical (LED-based control panel) devices are used to issue warnings to the driver in case dangerous conditions are detected, while a color monitor is mainly used as a debugging tool.

A mechanical device provides autonomous steering capabilities. It is composed of an electric stepping motor coupled to the steering column by means of a belt. The output fed by the vision system is used to turn the steering wheel to maintain the vehicle inside the lane or follow the leading vehicle.

### B. System Functionalities

Thanks to a control panel, the driver can select the level of system intervention. The following three driving modes are integrated.

1) *Manual Driving*: The system simply monitors and logs the driver's activity.

2) *Supervised Driving*: In case of danger, the system warns the driver with acoustic and optical signals. A LED row on the control panel encodes the vehicle position within the lane, while the right or left speakers warn in case the vehicle is approaching too much to the right or left lane marking, respectively.

3) *Automatic Steering*: The system maintains the full control of the vehicle's trajectory, and the two following functionalities can be selected:

**Road Following:** the automatic movement of the vehicle inside the lane. It is based on: *Lane Detection* (which includes the localization of the road, the determination of the relative position between the vehicle and the road, and the analysis of the vehicle's heading direction) and *Obstacle Detection* (which is mainly based on localizing possible generic obstacles on the vehicle's path).

**Platooning:** The automatic following of the preceding vehicle, that requires the localization and tracking of a target vehicle (*Vehicle Detection and Tracking*).

## III. OBSTACLE DETECTION

The Obstacle Detection functionality is aimed at the *localization of generic* objects that can obstruct the vehicle's path, without their complete *identification* or *recognition*. For this purpose a complete 3-D reconstruction is not required and a match with a given model suffices: the model represents the environment without obstacles, and any deviation from the model represents a potential obstacle.

### A. The Inverse Perspective Mapping

The Obstacle Detection functionality is based on the removal of the perspective effect through the Inverse Perspective Mapping (IPM) [13]. The IPM allows to remove the perspective effect from incoming images remapping each pixel toward a different position. It exploits the knowledge about the acquisition parameters (camera orientation, position, optics, etc) and the assumption of a flat road in front of the vehicle. The result is a new two-dimensional (2-D) array of pixels (the *remapped image*) that represents a bird's eye view of the road region in front of the vehicle [Fig. 2(c) shows the result of the application of IPM technique on the image Fig. 2(a)].

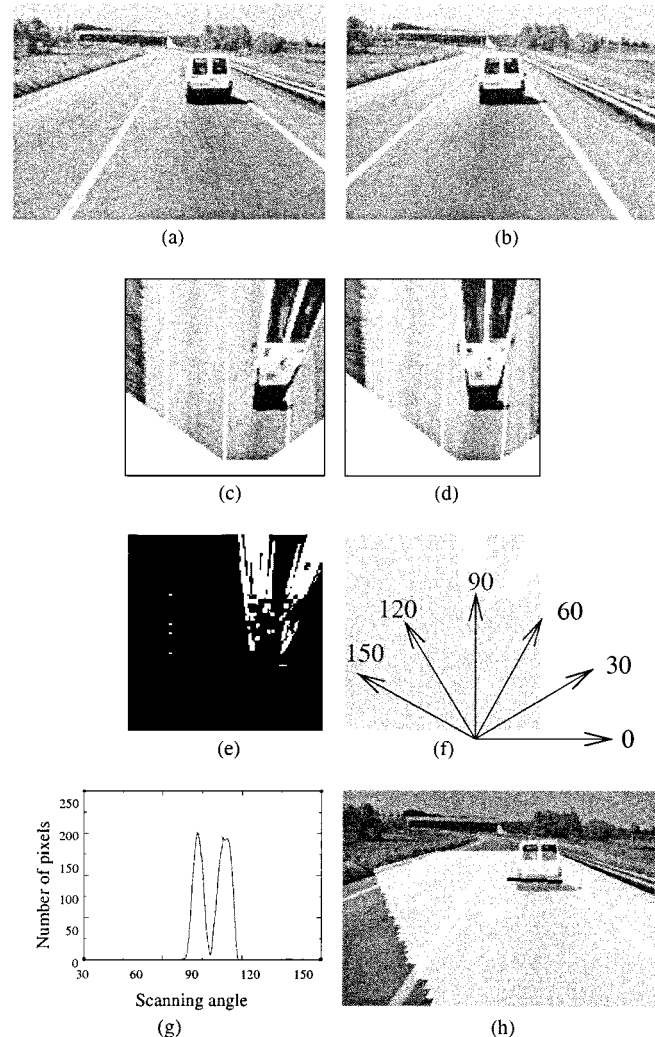


Fig. 2. Obstacle Detection: (a) the left stereo images; (b) the right stereo image; (c) and (d) the remapped images; (e) the difference image; (f) the angles of view overlapped with the difference image; (g) the polar histogram; and (h) the result of Obstacle Detection using a black marker superimposed on the original left image; the light-gray area represents the road region visible from both cameras.

### B. Obstacle Detection Processing Steps

The application of IPM to stereo images [12], [13] plays a strategic role for Obstacle Detection.

Assuming a *flat road*, the IPM is performed on both stereo images.<sup>1</sup> The flat road model is checked through a pixel-wise difference between the two remapped images: in correspondence to a *generic obstacle* in front of the vehicle, namely anything rising up from the road surface, the difference image features sufficiently large clusters of nonzero pixels that possess a particular shape. Due to the stereo cameras' different angles of view, an ideal homogeneous square obstacle produces two clusters of pixels with a triangular shape in the difference image, in correspondence to its vertical edges [13].

<sup>1</sup>An alternative solution is to warp the left or right image to the domain of the other one. Nevertheless, the Lane Detection functionality [12], not described in this work, relies on the image generated by the IPM. Moreover, the mapping of both images onto the ground eases the computation of object's distance as well as other features.



Fig. 3. Obstacle Detection. Result is shown with a black marking superimposed onto a brighter version of the image captured by the left camera; a black thin line limits the portion of the road seen by both cameras.

Unfortunately due to texture, irregular shape, and nonhomogeneous brightness of generic obstacles, in real cases the detection of the triangles becomes difficult. Nevertheless, in the difference image some clusters of pixels with a quasitriangular shape are anyway recognizable, even if they are not clearly disjointed. Moreover, in case two or more obstacles are present in the scene at the same time, more than two triangles appear in the difference image. A further problem is caused by partially visible obstacles which produce a single triangle.

The low-level portion of the process (see Fig. 2) is consequently reduced to the computation of the difference between the two remapped images, a threshold, and a morphological opening aimed at removing small-sized details in the thresholded image.

The following process is based on the localization of pairs of triangles in the difference image by means of a quantitative measurement of their shape and position [14]. It is divided into: computing a polar histogram for the detection of triangles, finding and joining the polar histogram's peaks to determine the angle of view under which obstacles are seen, and estimating the obstacle distance.

1) *Polar Histogram*: A polar histogram is used for the detection of triangles: it is computed scanning the difference image with respect to a point called *focus* and counting the number of over-threshold pixels for every straight line originating from the focus. It is important to note that the image area considered when building the polar histogram is not uniform along the scanning angle: under small angles, the considered sector is short, while for angles close to  $90^\circ$ , it gets longer. Therefore, the polar histogram's values are normalized applying a nonconstant threshold computed using the polar histogram of an image where all pixels are set. Finally, a low-pass filter is applied in order to decrease the influence of noise.

The polar histogram's focus is placed in the middle point between the projection of the two cameras onto the road plane; in this case the polar histogram presents an appreciable peak corre-

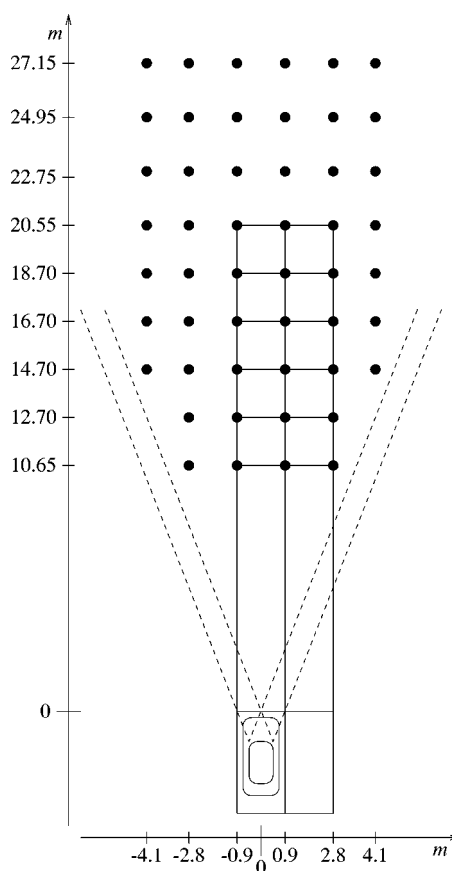


Fig. 4. Test-bed (black circles indicate the positions where obstacles have been placed).

sponding to each triangle [13]. Since the presence of an obstacle produces two disjointed triangles (corresponding to its edges) in the difference image, Obstacle Detection is limited to the search for pairs of adjacent peaks. The position of a peak in fact determines the angle of view under which the obstacle edge is seen.

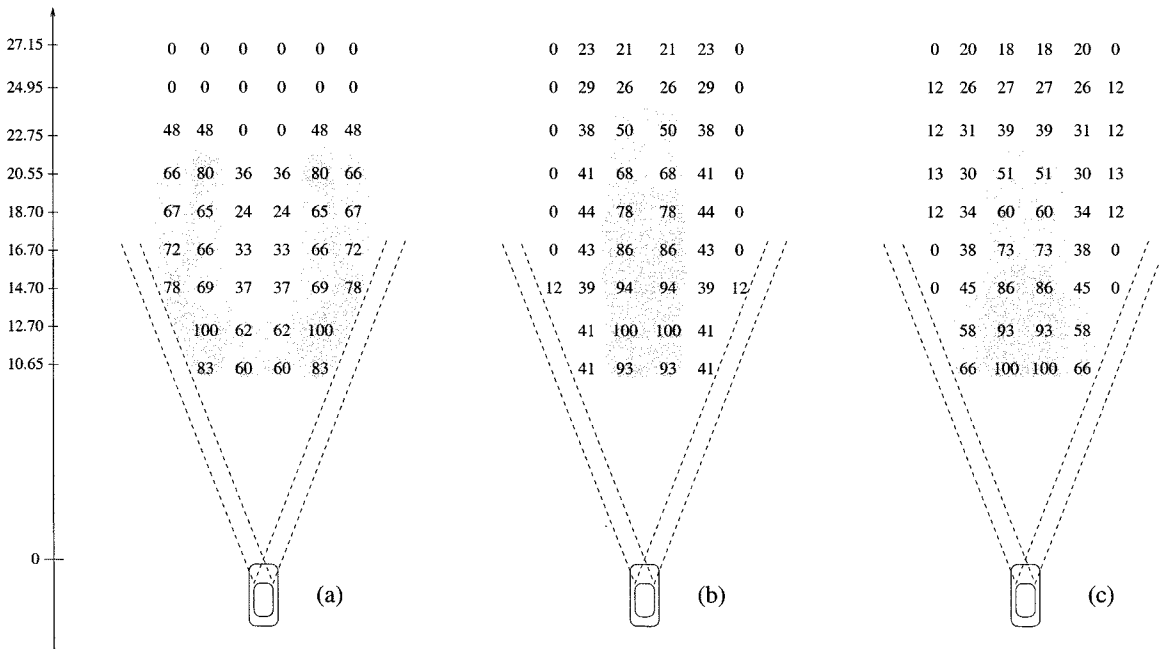


Fig. 5. Measured sensitivity in a 0–100 scale for three different kind of obstacles: (a) small and short obstacle; (b) large and tall obstacle; and (c) human shape.

Peaks may have different characteristics, such as amplitude, sharpness, or width. This depends on the obstacle distance, angle of view, and difference of brightness and texture between the background and the obstacle itself.

2) *Peaks Joining*: Two or more peaks can be joined according to different criteria. Starting from the analysis of a large number of different situations a criterion has been found, aimed to the grouping of peaks, that takes into account several characteristics such as the peaks amplitude and width, the area they subtend, as well as their distance [15]. Obviously, a partially visible obstacle produces a single peak that cannot be joined to any other.

The amplitude and width of peaks, as well as the interval between joined peaks, are used to determine the angle of view under which the whole obstacle is seen.

3) *Estimation of Obstacle Distance*: The difference image can also be used to estimate the obstacle distance. For each peak of the polar histogram a *radial histogram* is computed scanning a specific sector of the difference image. The width of this sector is determined from the width of the polar histogram peak [14]. The number of over-threshold pixels in the sector is computed as a function of the distance from the cameras and the result is normalized. The radial histogram is analyzed to detect the corners of triangles, which represent the contact points between obstacles and road plane, therefore allowing the determination of the obstacle distance through a simple threshold [13].

### C. Results

Concerning qualitative outcomes, Fig. 3 shows the results obtained in a number of different situations including multiple obstacles placed in different positions inside the stereo field of view. The result is displayed with black markings superimposed on a brighter version of the left image; markers encode both the obstacles' distance and width.

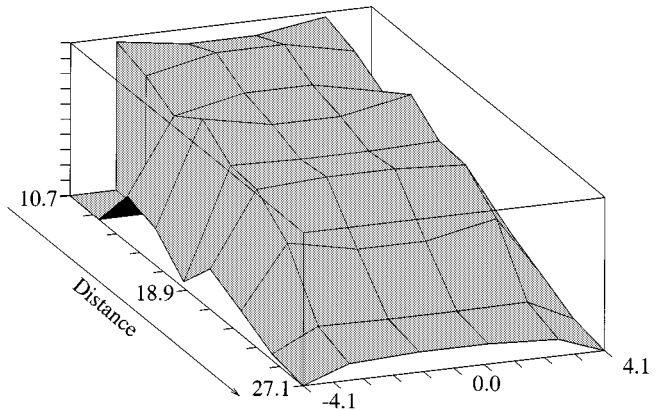


Fig. 6. Average values of the sensitivity for Obstacle Detection.

### D. Performance Analysis

Due to its fundamental importance, the Obstacle Detection module must be extremely robust and must reliably detect objects in a given distance range (i.e., in 100% of all cases). In order to evaluate the performance of the algorithm implemented on ARGO and determine possible enhancements, extensive tests have been carried out. Previous experiments [15] demonstrated that Obstacle Detection is robust to errors in vision system calibration (i.e., vehicle movements or deviations from the flat road hypothesis like the ones that should be expected in a highway/freeway scenario).

Nevertheless, an extensive test has been carried out for determining the sensitivity of Obstacle Detection to dimensions and position of obstacles.

1) *The Test-Bed*: Obstacles with different size and shape have been positioned in front of the vehicle at given distances and the sensitivity of the algorithm has been measured. The

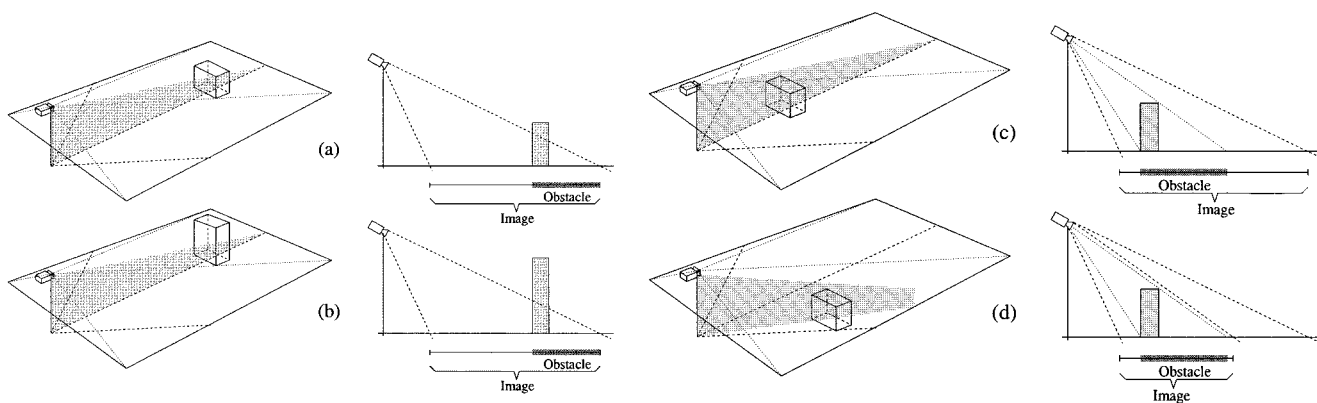


Fig. 7. Three-dimensional scene and projection of the obstacle on a linear profile of the image: (a) a small obstacle far from the camera; (b) a high obstacle far from the camera; (c) a small obstacle near the camera; and (d) a small obstacle near the camera but located on the right of the viewing region.

obstacle's characteristics that have been varied during the tests are the following:

- obstacle's position: ahead distance and lateral offset, ranging from 10 to 30 m for the distance perpendicular to the camera's stereo rig and from  $-4$  to  $4$  m for the lateral offset;
- obstacle's size: the tests included small obstacles (25 cm wide  $\times$  60 cm high) and large ones (50 cm wide  $\times$  90 cm high);
- obstacle's height: the range varied from 60 to 180 cm in height.

Moreover, sensitivity to human shapes has been tested.

During the tests, the following set-up and assumptions were used:

- The vehicle was standing still. Since noise is generally due to drifts in the cameras' calibration (generated by vehicle movements), this assumption permitted to remove this kind of noise.
- The obstacle's color has been selected to be homogeneous and different from the background.

Although many experiments were performed, this section reports on the tests made with the following three obstacles:

- small obstacle: 25 cm wide  $\times$  60 cm high;
- large obstacle: 50 cm wide  $\times$  90 cm high;
- human shape: 40 cm wide  $\times$  180 cm high.

The obstacles have been positioned on a grid, shown in Fig. 4.

2) *Obstacle Detection Sensitivity*: In order to determine the sensitivity ( $S$ ) to obstacles, the height of the polar histogram peak ( $H$ ) is analyzed and compared to the threshold ( $T$ ) used for the decision whether the peak is due to an obstacle or noise. In addition, the sensitivity is normalized using the height value ( $H_{\max}$ ) of the highest peak, namely

$$S = \begin{cases} 0 & \text{where } H < T \\ \frac{H}{H_{\max}} & \text{where } H \geq T \end{cases}$$

When two or more peaks are localized in the polar histogram, the highest is considered.

Since different illumination conditions can slightly affect the final result, several images have been acquired and processed

for each obstacle's position on the grid shown in Fig. 4. In case all the values were greater than the threshold their average was computed, otherwise a zero was taken.

Fig. 5 shows the results for three different obstacles: Fig. 5(a) shows a small sized obstacle; Fig. 5(b) shows a large and tall obstacle; and Fig. 5(c) shows a human shape. For each single obstacle, the values representing the sensitivity are scaled between 0 and 100, therefore they are not directly comparable.

However, in order to give an overview of the system's behavior, Fig. 6 graphically summarizes all the measurements: it has been computed as an average of all the tests performed on the different obstacles. It is clearly visible that the sensitivity to the presence of obstacles is high in the area right ahead of the vehicle (the cameras' angular aperture is nearly  $40^\circ$ ), and decreases—almost linearly—with the distance. The lateral regions have a lower sensitivity.

3) *Analysis of the Results*: The results obtained during the tests demonstrated that the sensitivity mainly depends on obstacle's height and position. Conversely the obstacle's width barely impacts on the sensitivity, affecting only the distance between the peaks of the polar histogram.

First of all, it is important to note that tall obstacles lying far from the camera share the same characteristics of short ones: this is due to the reduced region analyzed by the system, as it can be seen comparing Fig. 7(a) and (b). Therefore, the obstacle's height only impacts on the result when the obstacle is short enough to be fully visible by the cameras, as shown in Fig. 7(c). In this case, the sensitivity to obstacle's height is linear with the distance. This is clearly shown in Fig. 5: the closer the obstacle to the camera, the more reliable its detection.

Due to the variable threshold along the polar histogram's scanning angle, the system is much more sensitive to small obstacles when they lie on the sides of the viewing region. This behavior is explained by Fig. 7(d), which shows that in case of lateral obstacles, the considered area (sector) of the image is shorter than for the in-front analysis. Therefore, since the image profile is shorter, the projection of an obstacle covers a larger percentage of it, and thus the sensitivity to obstacles—and unfortunately also to noise—is higher in the peripheral (lateral) region. Fig. 5(a) confirms this behavior: a small obstacle is detected more reliably when it lies on the sides of the viewing area.

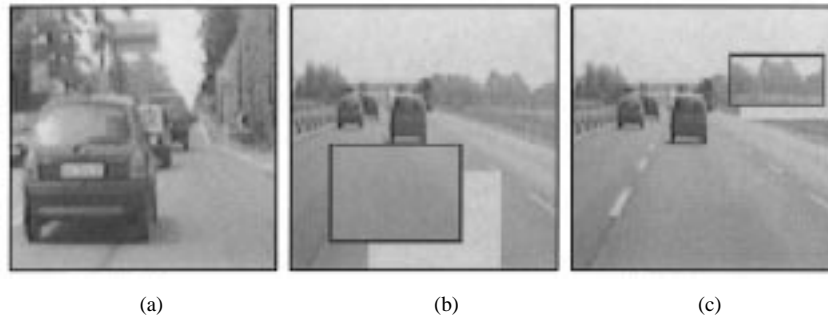


Fig. 8. Typical road scenes: (a) a strong sun reflection reduces the vehicle gray level symmetry; (b) a uniform area can be regarded as a highly symmetrical region; and (c) background symmetrical patterns.

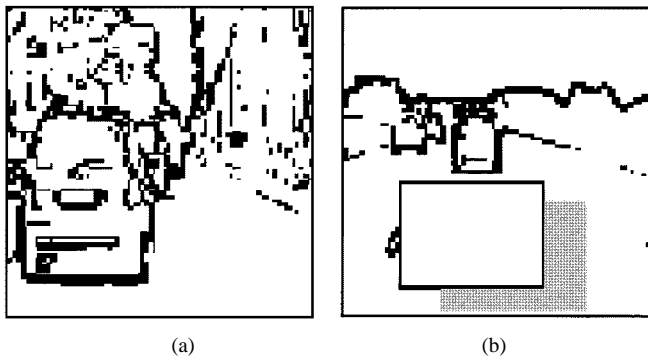


Fig. 9. Edges enforce the detection of real symmetries: (a) strong reflections have lower effects while (b) uniform areas are discarded since they do not present edges.

Since this approach is characterized by a low sensitivity to obstacles (and therefore the presence of noise becomes significant) in some areas, such as the region far away ahead of the vehicle, a new module is required to gain a better reliability and higher robustness in the detection in these areas. The next section presents the solution currently implemented on ARGO: a different algorithm which is limited to Vehicle Detection. It can be efficiently employed for Platooning due to its good sensitivity in the area far in front of the vehicle.

#### IV. VEHICLE DETECTION

The platooning functionality depends on a robust detection of the distance and speed of the preceding vehicle. Since Obstacle Detection does not generate sufficiently reliable results—in particular regarding obstacle *distance*,—a new functionality (Vehicle Detection) has been considered; the vehicle is localized and tracked using a single monocular image sequence; the correct distance is refined thanks to stereo vision.

The Vehicle Detection algorithm is based on the following considerations: a vehicle is generally symmetric, characterized by a rectangular bounding box which satisfies specific aspect ratio constraints, and placed in a specific region of the image. First an area of interest is identified on the basis of road position and perspective constraints. This area is searched for possible vertical symmetries; not only gray level symmetries are considered, but vertical and horizontal edges symmetries as well, in order to increase the detection robustness. Once the width and position of the symmetrical area have been detected, a new search begins, aimed at the detection of the two bottom corners

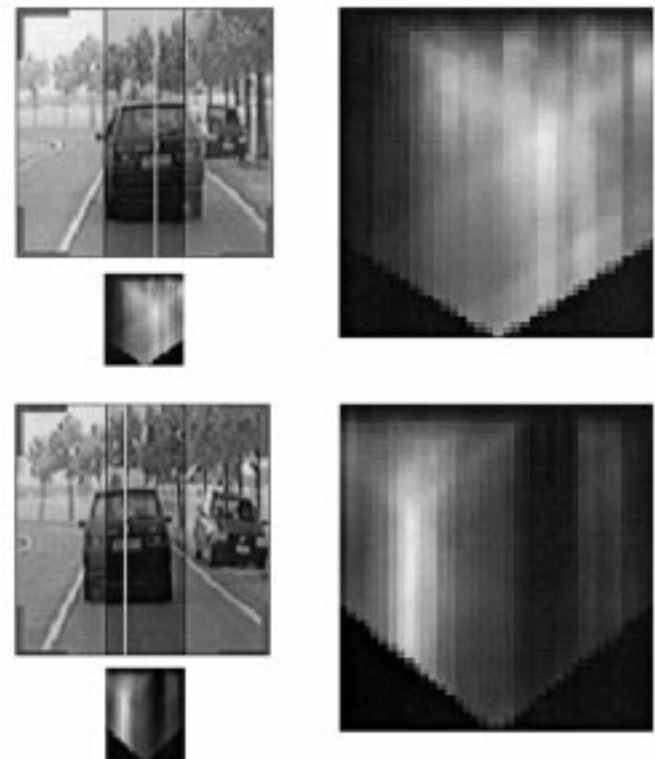


Fig. 10. Grey level symmetries: the two rightmost images show the enlarged symmetry maps encoding high symmetries with bright points.

of a rectangular bounding box. Finally, the top horizontal limit of the vehicle is searched for, and the preceding vehicle localized.

The tracking phase is performed through the maximization of the correlation between the portion of the image contained in the bounding box of the previous frame (partially stretched and reduced to take into account small size variations due to the increment or reduction of the relative distance) and the new frame.

##### A. Vehicle Detection Processing Steps

1) *Symmetry Detection*: In order to search for symmetrical features, the analysis of gray level images is not sufficient. Fig. 8 shows that strong reflections cause irregularities in vehicle symmetry, while uniform areas and background patterns may present highly correlated symmetries. In order to cope

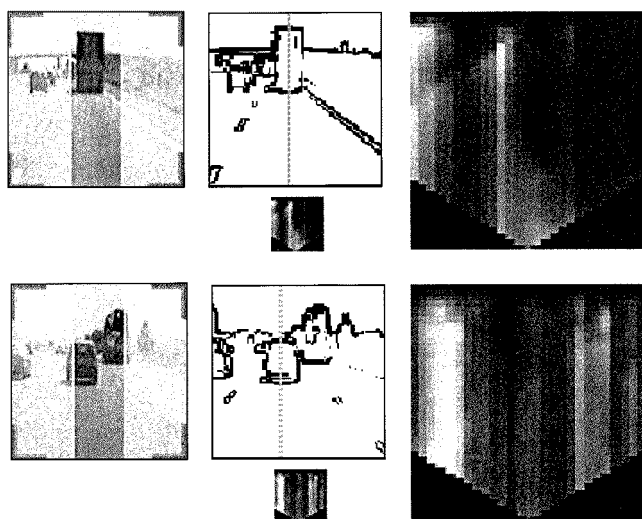


Fig. 11. Edge symmetries: the symmetries are computed on the binary images obtained after thresholding the gradient image.

with these problems, also symmetries in other domains are computed.

To get rid of reflections and uniform areas, vertical and horizontal edges are extracted and thresholded, and symmetries are computed in these domains as well. Fig. 9 shows that, although a strong reflection is present on the left side of the vehicle, edges are anyway visible and can be used to extract symmetries; moreover, in uniform areas no edges are extracted and therefore no symmetries are detected. Fig. 10 shows two examples in which gray level symmetries alone can be successful for vehicle detection, while Fig. 11 shows the result of edge symmetry.

For each image, the search area is shown in dark gray and the resulting vertical axis is superimposed. For each image its symmetry map is also depicted both in its original size and—on the right—zoomed for better viewing. Bright points encode the presence of high symmetries. The 2-D symmetry maps are computed for different values of the axis' horizontal position within the grey area in the original image (horizontal axis) and the horizontal width of the symmetry area (vertical axis). The lower triangular shape is due to the limitation in scanning large horizontal windows for peripheral vertical axes.

Similarly, the analysis of symmetries of horizontal and vertical edges produces other symmetry maps, which, with specific coefficients detected experimentally which depend on the vision system set-up, can be combined with the previous ones to form a single symmetry map. Fig. 12 shows all symmetry maps and the final one, that allows to detect the vehicle.

2) *Bounding Box Detection*: After the localization of the symmetry, the symmetrical region in the edge image is checked for the presence of two corners representing the bottom of the bounding box around the vehicle. A traditional pattern matching technique is used. Moreover, perspective constraints as well as size constraints are used to speed-up the search. Fig. 13 shows possible and impossible bottom parts of the bounding box, while Fig. 14 presents the results of the lower corners detection.

This process is followed by the detection of the top part of the bounding box, which is looked for in a specific region whose

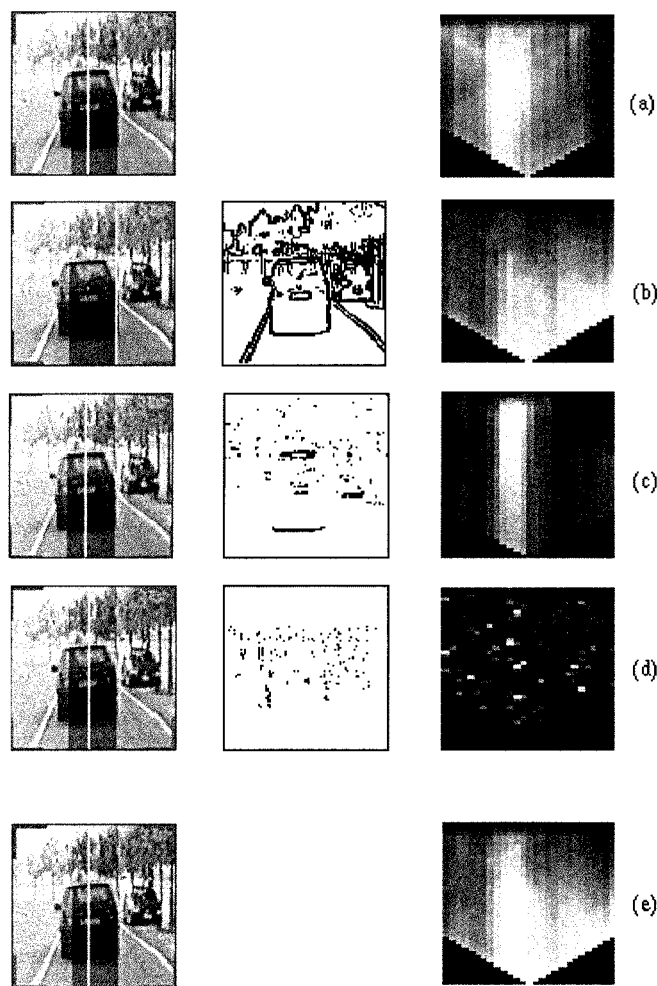


Fig. 12. Computing the resulting symmetry: (a) grey-level symmetry; (b) edge symmetry; (c) horizontal edges symmetry; (d) vertical edges symmetry; and (e) total symmetry. For each row, the resulting symmetry axis is superimposed onto the leftmost original image.



Fig. 13. Detection of the lower part of the bounding box. (a) Correct position and size, taking into consideration perspective constraints and knowledge on the acquisition system setup, as well as typical vehicles' (b) Incorrect bounding boxes.

location is again determined by perspective and size constraints. Fig. 15 shows the search area.

3) *Backtracking*: Sometimes it may happen that in correspondence to the symmetry maximum no correct bounding boxes exist. Therefore, a backtracking approach is used: the symmetry map is again scanned for the next local maximum and a new search for a bounding box is performed. Fig. 16 shows a situation in which the first symmetry maximum,



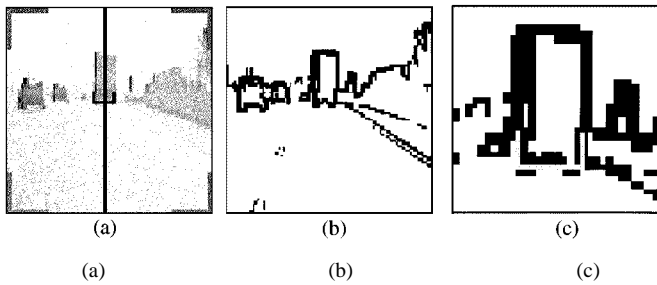


Fig. 14. Detection of the lower part of the bounding box: (a) original image with superimposed results; (b) edges; and (c) localization of the two lower corners.



Fig. 15. Search area for the upper part of the bounding box is shown in dark gray. It takes into account knowledge about the typical vehicles' aspect ratio.

generated by a building, does not lead to a correct bounding box; on the other hand, the second maximum leads to the correct detection of the vehicle.

4) *Distance Refinement*: The distance to the leading vehicle is computed with the knowledge of the camera calibration. Unfortunately, it may assume wrong values since it may happen that the lower part of the vehicle is not correctly detected. 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, which is of importance for the platooning functionality, an adjustment step is mandatory: it is performed taking advantage of stereo techniques. Starting from the distance value estimated from the left image, a portion of the right image is searched for a pattern similar to the one enclosed into the bounding box.

This step relies on the following assumptions:

- the rear side of the vehicle is approximated as a vertical plane;
- luminance differences in the vehicle pattern, caused by light reflections, are negligible in the two stereo views.

Since the stereo set-up is known, once the same pattern enclosed into the bounding box is detected on the right image, a simple triangulation allows to measure the vehicle distance: the offset of the bounding boxes containing the vehicle, measured in both images, is used to compute the vehicle distance.

Besides being simpler than traditional stereo-based techniques, this approach has the following advantages:

- it only requires one triangulation since the computation of the vehicle distance is the only final goal;

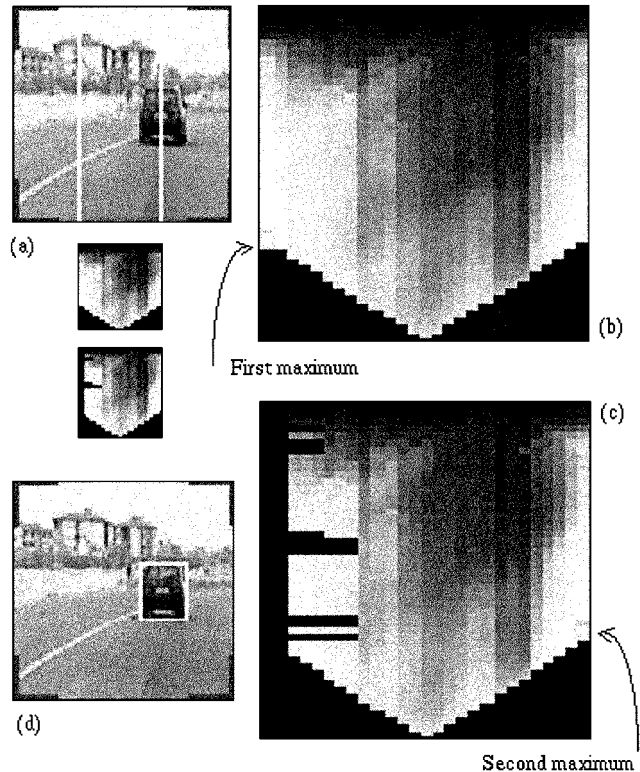


Fig. 16. Case in which the background symmetry is higher than the vehicle symmetry: (a) original image; (b) first symmetry map; (c) second symmetry map after the backtracking process has removed the peak near the maximum; and (d) final bounding box detection.

- errors are reduced to a minimum since the triangulation refers to a large and complex pattern whose identification is fairly easy;
- since not only the search pattern is known, but an estimate of the vehicle distance as well, the search is performed only in a reduced region of the image and therefore this step is not as computation intensive as traditional stereo techniques.

Fig. 17 shows the steps used for distance refinement: Fig. 17(b) shows the incorrect result of the detection step, Fig. 17(c) shows that using a null offset the vehicle in the two images does not overlap, while figure Fig. 17(d) shows that a specific offset brings the two rears to a perfect correspondence.

## B. Results

Fig. 18 shows some qualitative results of Vehicle Detection in different situations: the preceding vehicle is correctly detected at different distances, even on complex scenes.

## V. COMPUTATIONAL PERFORMANCE

Table I shows the timing performance; since Obstacle and Lane Detection (the latter is not described in this work) share the removal of the perspective effect, the timings for IPM are separated from the others. In addition, due to the different computational burden of Vehicle Detection when looking for a vehicle or tracking an already found one, two distinct timings for Vehicle Detection and Tracking are shown.



Fig. 17. Distance refinement: (a) left and right stereo images; (b) incorrect result of the detection step (the lower part of the bounding box indicates a wrong distance); (c) superimposition of stereo images with a null offset; and (d) superimposition of correctly shifted stereo images.

The acquisition adapter installed on the ARGO system is able to continuously capture images into a circular buffer in main memory, therefore, not requiring a synchronization with the processing (see Fig. 19).

When all the three functionalities are turned on, the system can work up to a 45 Hz rate.

## VI. VEHICLE CONTROL

This section addresses the problem of automatic steering which has recently gained considerable attention from both the theoretical [16], [17] and experimental side [18], [19].

Roughly speaking this problem is centered on finding a satisfactory law for the command of the steering wheel. Many works have been reported in the literature [20], [21], and various steering control designs were proposed for systems in which the

sensing is performed with nonvisual devices (e.g., guiding wire, microwave radars, etc.).

On the other hand, a visual servoing paradigm was proposed by Epiou *et al.* [22] by considering a simple omnidirectional mobile robot. Neural networks were adopted and subsequently developed at CMU [23], [24]. A comparative survey on various vision-based control strategies for autonomous vehicles can be found in the paper by Taylor *et al.* [19].

The following paragraph presents a gain scheduled proportional controller currently implemented on the ARGO vehicle. By using a feedback supervisor this control law can be adopted to perform both Path Following and Platooning. A simple proportional control law was previously examined by Özgüner *et al.* [25] for the Road Following functionality solely.

### A. Gain Scheduled Proportional Controller

The controller currently adopted for the ARGO vehicle was initially designed and optimized for the Road Following task [12]. Minor changes have been introduced to implement also the Platooning functionality.

The basic control scheme is visible in Fig. 20. The command steering angle  $\delta$  is obtained with a variable gain proportional controller. The vision based system reconstructs the road environment and the supervisor uses the results to select the most appropriate gain for the proportional controller and estimate the error signal. Initially, the offset  $e$  existing between the vehicle heading and the desired path is computed at the look-ahead distance  $L$  (see Fig. 21). The estimated signal  $e$  is inherently noisy so that it cannot be directly supplied to the proportional controller. To reduce the disturbances,  $e$  is preliminary filtered with a moving average filter. The look-ahead distance is variable and depends on the vehicle speed; more precisely,  $L$  is obtained according to the following expression:

$$L(v) := \begin{cases} L_{\min} & \text{if } v < v_{\min} \\ vt_l & \text{if } v_{\min} \leq v \leq v_{\max} \\ L_{\max} & \text{if } v > v_{\max} \end{cases} \quad (1)$$

where  $L_{\min} = v_{\min}t_l$  and  $L_{\max} = v_{\max}t_l$  indicate the minimum and maximum look-ahead distance, respectively,  $t_l$  is the look-ahead time, and  $v$  is the vehicle speed.  $L(v)$  is a continuous function because  $v_{\min}$  and  $v_{\max}$  are chosen according to the relations  $L_{\min}/v_{\min} = L_{\max}/v_{\max} = t_l$ . Basically,  $L$  is proportional to the vehicle speed but it is saturated not to exceed the bounds imposed by the vision system, i.e.,  $L(v) \in [L_{\min}, L_{\max}]$ . The choice of  $L$  influences the behavior of the controller. It has been demonstrated [19] that, as  $v$  increases, the damping factor of the closed loop system gets worse and can be improved, under certain limits, by increasing the look-ahead distance. For the ARGO vehicle, the supervisor uses the parameters reported in Table II.

To further improve the performances of the closed loop system a gain scheduling technique has been adopted for the proportional controller. Specifically, controller gain  $K$  inversely depends on the velocity  $v$  according to

$$K(v) := \begin{cases} K_{\max} & \text{if } v < v^* \\ K_A/v & \text{if } v^* \leq v. \end{cases} \quad (2)$$



Fig. 18. Vehicle Detection: the images show the search area and the detected vehicle with black markings superimposed onto a brighter version of the original image.

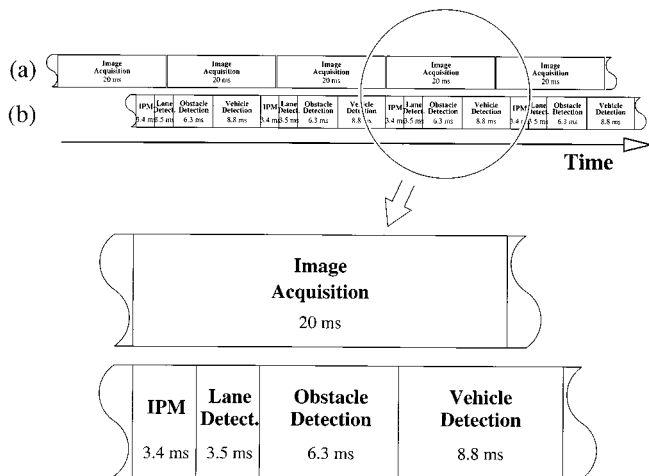


Fig. 19. Image acquisition (a) and processing (b) performance.

TABLE I  
TIMINGS OF PROCESSING STEPS

	Pentium 200MMX	Pentium II 450MMX	Speedup
IPM	9.9 ms	3.4 ms	2.9
Lane Detection	14.4 ms	3.5 ms	4.1
Obstacle Detection	17.5 ms	6.3 ms	2.7
Vehicle Tracking	24.8 ms	8.8 ms	2.8
Vehicle Detection	47.6 ms	19.9 ms	2.4

If the velocity becomes smaller than  $v^*$ , the proportional gain is upper bounded by  $K_{\max}$  (for ARGO  $v^* = 2.777 \text{ ms}^{-1} = 10 \text{ km/h}$ ).  $K(v)$  is continuous because  $K_A$  must satisfy the equation  $K_{\max} = K_A/v^*$ . The parameter  $K_A$  (and consequently  $K_{\max}$ ) has been set by means of a series of experiments on ARGO.

The controller sampling time is imposed by the vision system (it is given by the refresh rate of the cameras) and is equal to 0.02 s (50 Hz).

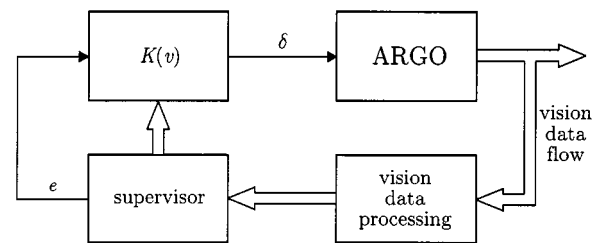


Fig. 20. Control scheme with the gain scheduled proportional controller.

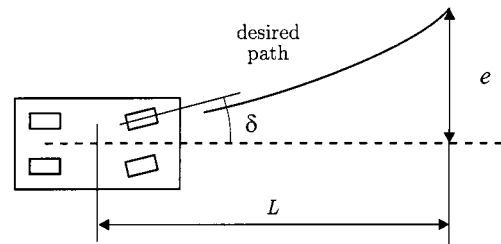


Fig. 21. Offset from the desired path, estimated by the vision system.

TABLE II  
LOOK-AHEAD DISTANCE PARAMETERS

$v_{\min}$	$= 8.33 \text{ ms}^{-1}$ (30 km/h)
$v_{\max}$	$= 22.22 \text{ ms}^{-1}$ (80 km/h)
$t_l$	$= 1.5 \text{ s}$
$L_{\min}$	$= 12.5 \text{ m}$
$L_{\max}$	$= 33.33 \text{ m}$

The control strategy adopted for Platooning takes advantage of the previously defined control scheme (see Fig. 20). The main and crucial difference with respect to the path-following functionality is on the supervisor estimation of the offset error  $e$ . When the Platooning functionality is activated, the target point is centered on the preceding vehicle so that the target look-ahead

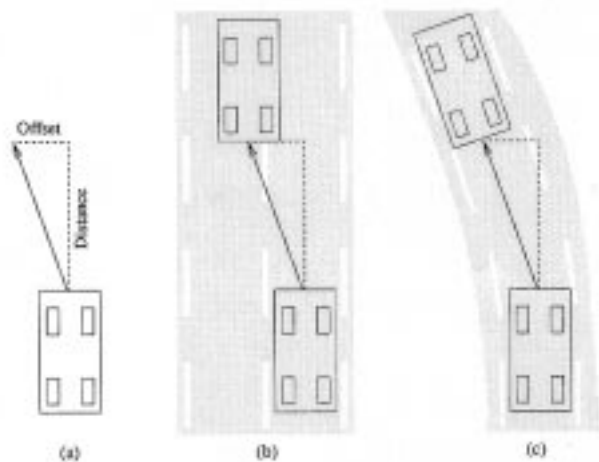


Fig. 22. Evaluation of the proper error signal for a Platooning application.

distance  $L'$  is neither constant nor the most appropriate for the current velocity (see Fig. 22). Obviously, the use of this look-ahead distance  $L'$  and the corresponding target offset error  $e'$  could degrade the performance of the Platooning functionality. The efficiency of the Platooning control algorithm is recovered by scaling the tracking error  $e'$  measured at  $L'$  to an estimated offset error  $e(v)$  given through a *virtual* target point placed at the appropriate look-ahead distance  $L(v)$  [cf. (1)]

$$e(v) := \frac{L(v)}{L'} e'. \quad (3)$$

Indeed, (3) provides the suitable  $e(v)$  regardless of the actual distance of the target vehicle, i.e.,  $L'$  can be greater or less than  $L(v)$  indifferently.

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, the Obstacle Detection module of the ARGO prototype was discussed. Since the robust localization of obstacles and other vehicles is a basic prerequisite for both *Road Following* and *Platooning* functionalities, also an experimental performance evaluation test was performed with the aim of determining the main bottlenecks and devising possible enhancements.

The test results indicate that the Obstacle Detection functionality is weak in some areas, such as the lateral or far ones. This result suggested the development of the Vehicle Detection functionality, that relies on shape information and is now integrated on ARGO.

Preliminary experimental results demonstrated that preceding vehicles are correctly detected at different distances, even on complex scenes. Therefore, the simple control algorithm described in Section VI, which had proved to be sufficiently robust for Lane Detection [26], has been applied to Platooning. However, the higher complexity of this task requires a more sophisticated approach. In fact, the leading vehicle position is used by the Platooning functionality to automatically *follow* the vehicle ahead, but it is not enough to thoroughly *reproduce its trajectory*. As Fig. 23 shows, not only the current position of the leading vehicle is required, but its trajectory must be reconstructed as well, since the lateral offset

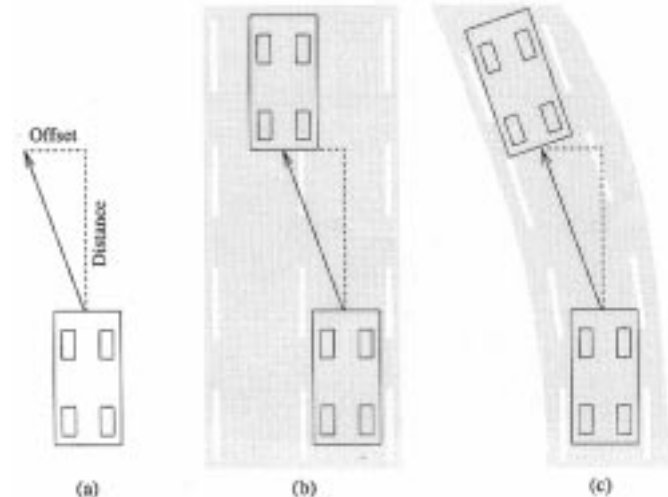


Fig. 23. Knowledge of the offset is not enough: the offset measurement shown in (a) may correspond to the two following situations: (b) overtaking on a straight road and (c) driving on a curved road, requiring the generation of two different trajectories.

between the two vehicles must be reduced to zero according to different rules depending on the situation.

New control strategies and the integration of the results of the Lane Detection module are now under study and will be discussed in a future paper. Their integration on the ARGO vehicle will lead to:

- 1) superior road following with smooth cruising and
- 2) highly-flexible functionality.

In particular, flexibility can be simply obtained by modifying the supervisor strategy in order to perform, e.g., lane changing, lane inserting, Platooning, and even car parking maneuvers.

## REFERENCES

- [1] M. Betke, E. Haritaoglu, and L. Davis, "Multiple vehicle detection and tracking in hard real-time," in *Proc. IEEE Intelligent Vehicles Symp.*, Tokyo, Japan, Sept. 1996, pp. 351–356.
- [2] S. Kyo, T. Koga, K. Sakurai, and S. Okazaki, "A robust vehicle detecting and tracking system for wet weather conditions using the IMAP-VISION image processing board," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, Tokyo, Japan, Oct. 1999, pp. 423–428.
- [3] S. Denasi and G. Quaglia, "Early obstacle detection using region segmentation and model-based edge grouping," in *Proc. IEEE Intelligent Vehicles Symp.*, Stuttgart, Germany, Oct. 1998, pp. 257–262.
- [4] M. Lützelner and E. D. Dickmanns, "Road recognition with MarVEye," in *Proc. IEEE Intelligent Vehicles Symp.*, Stuttgart, Germany, Oct. 1998, pp. 341–346.
- [5] W. Kruger, W. Enkelmann, and S. Rossle, "Real-time estimation and tracking of optical flow vectors for obstacle detection," in *Proc. IEEE Intelligent Vehicles Symp.*, Detroit, MI, Sept. 1995, pp. 304–309.
- [6] T. Suzuki and T. Kanade, "Measurement of vehicle motion and orientation using optical flow," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, Tokyo, Japan, Oct. 1999, pp. 25–30.
- [7] U. Franke, D. Gavrilu, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler, "Autonomous driving goes downtown," *IEEE Intell. Syst.*, vol. 13, pp. 40–48, Nov.–Dec. 1998.
- [8] D. Koller, J. Malik, Q.-T. Luong, and J. Weber, "An integrated stereo-based approach to automatic vehicle guidance," in *Proc. 5th Int. Conf. Computer Vision*, Boston, MA, 1995, pp. 12–20.
- [9] T. Williamson and C. Thorpe, "Detection of small obstacles at long range using multibaseline stereo," in *Proc. IEEE Intelligent Vehicles Symp.*, Stuttgart, Germany, Oct. 1998, pp. 311–316.
- [10] S. M. Smith and J. M. Brady, "ASSET-2: Real-time motion segmentation and shape tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 814–829, 1995.

- [11] S. M. Smith, "ASSET-2: Real-time motion segmentation and object tracking," in *Real Time Imaging Journal*. New York: Academic, Feb. 1998, vol. 4, pp. 21–40.
- [12] A. Broggi, M. Bertozzi, A. Fascioli, and G. Conte, *Automatic Vehicle Guidance: The Experience of the ARGO Vehicle*. Singapore: World Scientific, Apr. 1999.
- [13] M. Bertozzi, A. Broggi, and A. Fascioli, "Stereo inverse perspective mapping: Theory and applications," *Image Vis. Comput.*, vol. 8, no. 16, pp. 585–590, 1998.
- [14] A. Fascioli, "Vision-based automatic vehicle guidance: Development and test of a prototype," Ph.D. dissertation, Dipartimento di Ingegneria dell'Informazione, Università di Parma, Italy, Jan. 2000.
- [15] M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Processing*, vol. 7, pp. 62–81, Jan. 1998.
- [16] R. Frezza, S. Soatto, and G. Picci, "Visual path following by recursive spline updating," in *Proc. 36th Conf. Decision and Control*, San Diego, CA, Dec. 1997, pp. 1130–1134.
- [17] Y. Ma, J. Košecák, and S. Sastry, "Vision guided navigation for a non-holonomic mobile robot," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 521–536, June 1999.
- [18] C. Hatipoğlu, K. Redmill, and Ü. Özgüner, "Steering and lane change: A working system," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, Boston, MA, Nov. 1997.
- [19] C. J. Taylor, J. Košecák, R. Blasi, and J. Malik, "A comparative study of vision-based lateral control strategies for autonomous highway driving," *Int. J. Robot. Res.*, vol. 18, pp. 442–453, May 1999.
- [20] T. Hessburg and M. Tomizuka, "Fuzzy logic control for lateral vehicle guidance," *IEEE Control Syst. Mag.*, vol. 14, pp. 55–63, Aug. 1994.
- [21] R. H. Byrne, C. T. Abdallah, and P. Dorato, "Experimental results in robust lateral control of highway vehicles," *IEEE Contr. Syst. Technol.*, vol. 18, no. 2, pp. 70–76, 1998.
- [22] B. Epiau, F. Chaumette, and P. Rives, "A new approach to visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 8, pp. 313–326, 1992.
- [23] D. A. Pomerleau, "Progress in neural network-based vision for autonomous robot driving," in *Proc. IEEE Intelligent Vehicles Symp.*, Detroit, MI, Oct. 1992, pp. 391–396.
- [24] T. M. Jochem, D. A. Pomerleau, and C. E. Thorpe, "MANIAC: A next generation neurally based autonomous road follower," in *Proc. 3rd Int. Conf. Intelligent Autonomous Systems*, Pittsburgh, PA, Feb. 1993.
- [25] Ü. Özgüner, K. A. Ünyelioglu, and C. Hatipoğlu, "An analytical study of vehicle steering control," in *Proc. IEEE Conf. Control Applications*, Washington, DC, 1995, pp. 125–130.
- [26] A. Broggi, M. Bertozzi, and A. Fascioli, "ARGO and the MilleMiglia in automatico tour," *IEEE Intell. Syst.*, vol. 14, pp. 55–64, Jan.–Feb. 1999.



**Alberto Broggi (S'90)** received the Dr.Eng. (Masters) degree in electronic engineering in 1990 and the Ph.D. degree in information technology in 1994, both from the Università di Parma, Italy.

From 1994 to 1998, he was a Full Researcher at the Dipartimento di Ingegneria dell'Informazione, Università di Parma, Italy. Since 1998, he has been Associate Professor of Artificial Intelligence at the Dipartimento di Informatica e Sistemistica, Università di Pavia, Italy. His research interests include real-time computer vision approaches for the navigation of un-

manned vehicles, and the development of low-cost computer systems to be used on autonomous agents. He is the Coordinator of the ARGO project, with the aim of designing, developing, and testing the ARGO autonomous prototype vehicle, equipped with special active safety features and enhanced driving capabilities. He is author of more than 100 refereed publications in international journals, book chapters, and conference proceedings. He is actively involved in the organization of scientific events, and is on the Editorial Board and Program Committee of many international journals and conferences and has been invited to act as Guest Editor of journals and magazines theme issues on topics related to intelligent vehicles, computer vision application, and computer architectures for real-time image processing.

Prof. Broggi is the Newsletter Editor and Member of the Conference and Publication Committees of the IEEE Intelligent Transportation Systems Council and is the Program Chair of the IEEE Intelligent Vehicles Symposium 2000, Detroit, MI <http://WWW.CE.UniPR.IT/people/broggi>



**Massimo Bertozzi (M'95)** received the Dr.Eng. (Masters) degree in electronic engineering and the Ph.D. degree in information technology, both from the Università di Parma, Italy, in 1994 and 1997, respectively. His thesis was on the implementation of simulation of Petri nets on the CM-2 massive parallel architecture.

From 1994 to 1997, he chaired the local IEEE student branch. During this period, his research interests focused mainly on the application of image processing to real-time systems and to vehicle guidance, the optimization of machine code at assembly level, and parallel and distributed computing. He is currently a Researcher in the Dipartimento di Ingegneria dell'Informazione, Università di Parma.

Dr. Bertozzi is a Member of IAPR.



**Alessandra Fascioli (M'97)** received the Dr.Eng. (Masters) degree in electronic engineering in 1996 and the Ph.D. degree in information technology in 2000 from the Università di Parma, Italy. Her thesis was on stereo vision-based obstacle localization in automotive environments.

She is currently a Temporary Researcher at the University of Parma. Her research interests focus on real-time computer vision and computer architectures for automatic vehicle guidance. She is also interested in image processing techniques based on the Mathematical Morphology computational model.

Dr. Fascioli chaired the University of Parma IEEE Student Branch from 1998 to 2000. She is also a member of AI\*IA and IAPR.



**Corrado Guarino Lo Bianco** was born in Sassari, Italy, on May 3, 1964. He received the Dr.Eng. degree (with honors) in electronic engineering and the Ph.D. degree in control system engineering from the University of Bologna, Italy, in 1989 and 1994, respectively.

Currently, he is with the Computer Engineering Department (CE) of the University of Parma as a Research Associate on System Engineering. He is involved in research on variable reluctance motors, power devices thermal analysis, smooth profile generation for motion control, robust control design via semi-infinite optimization, genetic algorithms, and interval analysis.



**Aurelio Piazza (M'93)** received the Laurea degree in nuclear engineering in 1982 and the Ph.D. degree in system engineering in 1987, both from the University of Bologna, Italy.

From 1990 to 1992, he was Research Associate in System Theory, D.E.I.S., University of Bologna. Since November 1992, he has been Associate Professor of Automatic Control, Dipartimento di Ingegneria dell'Informazione, University of Parma. His main research interests are in system and control theory and related engineering applications. His

recent research activities have focused on methods of global optimization applied to robotics and control problems and on dynamic inversion techniques for vision-based automatic steering and for the design of robust control systems.

Dr. Piazza is a Member of IFAC and SIAM.