LLIGENT
CLE
NOLOGIES

igation devices and cruise control are already being
d into modern vehicles. Manufacturers are now racing
o 'smart' cruise control, on-vehicle driver information
collision avoidance systems, vision enhancement and
iness diagnostics systems.

t Vehicle Technologies covers this growing field in
e engineering, from intelligent sensors to intelligent
tems. Undergraduates studying Automotive Engineering
a useful resource, as will practising engineers in intelligent
d automation, robotics and communications.

d with practical examples and applications

th treatment written in a text-book style
than a theoretical specialist text style)

nts ways to design and develop intelligent
e solutions

ational author team consists of **Dr L Vlacic**, based at
versity, Australia, **Dr M Parent** at INRIA, France, and
F Harashima, at Tokyo Metropolitan Institute of
Japan.

KS IN THE BUTTERWORTH-HEINEMANN
E ENGINEERING SERIES:

Electric/Hybrid Vehicle Design
d Fenton 0 7506 5092 3

le Structures
tson and Serpento 0 7506 5134 2

T T E R W O R T H
I N E M A N N
om

ISBN 0-7506-5093-1

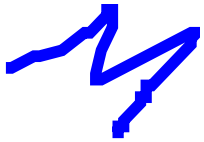
9 780750 650939

INTELLIGENT VEHICLE TECHNOLOGIES VLACIC • PARENT • HARASHIMA BH

L VLACIC
M PARENT
F HARASHIMA

BH

INTELLIGENT VEHICLE TECHNOLOGIES



14

ARGO prototype vehicle

Alberto Broggi
Dipartimento di Informatica e Sistemistica, University of Pavia, Italy

and

Massimo Bertozzi, Gianni Conte, and Alessandra Fascioli
Dipartimento di Ingegneria dell'Informazione, University of Parma, Italy

This chapter presents the experience of the ARGO project. It started in 1996 at the University of Parma, based on the previous experience within the European PROMETHEUS Project. In 1997 the ARGO prototype vehicle (described in Section 14.3) was set up with sensors and actuators, and the first version of the GOLD software system – able to locate one lane marking and generic obstacles on the vehicle's path – was installed. In June 1998 the vehicle underwent a major test (the *MilleMiglia in Automatico*, a 2000 km tour on Italian highways discussed in Section 14.4) in order to test the complete equipment. The analysis of this test enabled the improvement of the system. Section 14.2 presents the current implementation of the GOLD system, featured by enhanced lane detection abilities and extended obstacle detection abilities, such as the detection of leading vehicles and pedestrians.

14.1 Introduction: the ARGO project

The main target of the ARGO project is the development of an active safety system with the ability to act also as an automatic pilot for a standard road vehicle.

In order to achieve autonomous driving capabilities on the existing road network with no need for specific infrastructures, a robust perception of the environment is essential. 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 and must be lower than a safety threshold. For this reason in the implementation of the ARGO vehicle only the use of passive sensors, namely *cameras*, has been considered.

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 operative costs, which must not exceed a certain threshold in order not to interfere

with the vehicle performance. Therefore low-cost devices have been preferred, both for the image acquisition and the processing: the prototype installed on ARGO is based on *cheap cameras* and a *commercial PC*.

Section 14.2 presents the main functionalities integrated on the ARGO vehicle:

- lane detection and tracking
- obstacle detection
- vehicle detection and tracking.

A further functionality, pedestrian detection – needed for urban driving – is currently under development.

14.2 The GOLD system

GOLD is the acronym used to refer to the software that provides ARGO with autonomous capabilities. It stands for Generic Obstacles and Lane Detection since these were the two functionalities originally developed. Currently it integrates two other functionalities: vehicle detection and the new pedestrian detection, which is under development.

14.2.1 The inverse perspective mapping

The lane detection and obstacle detection functionalities share the same underlying approach: the removal of the perspective effect obtained through the inverse perspective mapping (IPM) (Bertozzi and Broggi, 1998; Broggi *et al.*, 1999).

The IPM is a well-established technique that allows the perspective effect to be removed when the acquisition parameters (camera position, orientation, optics etc.) are completely known and when a piece of information about the road is given, such as a *flat road hypothesis*. The procedure aimed at removing the perspective effect resamples the incoming image, remapping each pixel toward a different position and producing a new two-dimensional array of pixels. The so-obtained remapped image represents a top view of the road region in front of the vehicle, as it were observed from a significant height.

Figures 14.1(a) and (b) show an image acquired by ARGO's vision system and the corresponding remapped image respectively.

14.2.2 Lane detection

Lane detection functionality is divided in two parts: a lower-level part, which, starting from iconic representations of the incoming images, produces new transformed representations using the same data structure (array of pixels), and a higher-level one, which analyses the outcome of the preceding step and produces a symbolic representation of the scene.

Low- and medium-level processing for lane detection

Lane detection is performed assuming that a road marking in the remapped image is represented by a quasi-vertical bright line of constant width on a darker background

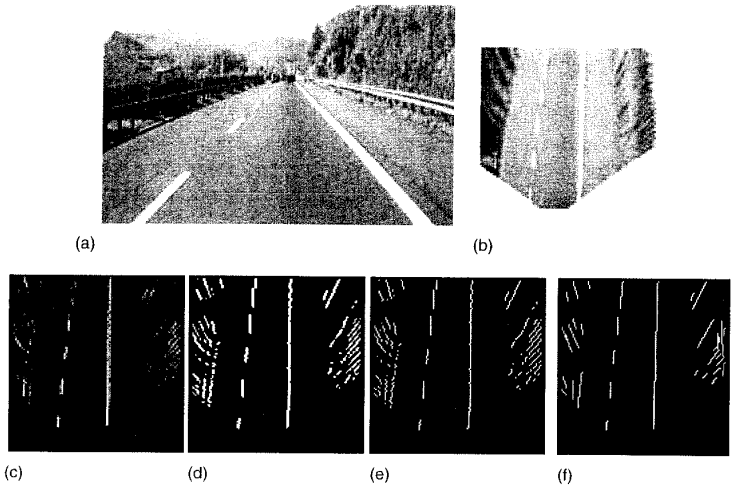


Fig. 14.1 The sequence of images produced by the low-level lane detection phase: (a) original; (b) remapped; (c) filtered; (d) enhanced; (e) binarized; (f) polylines.

(the road). Consequently, the pixels belonging to a road marking feature a higher brightness value than their left and right neighbours at a given horizontal distance.

The first phase of road markings detection is therefore based on a line-wise determination of horizontal black-white-black transitions, while the following medium-level process is aimed at extracting information and reconstructing road geometry.

Feature extraction The brightness value $b(x, y)$ of a generic pixel belonging to the remapped image is compared to its horizontal left and right neighbours at a distance m: $b(x, y - m)$ and $b(x, y + m)$, with $m \geq 1$.

A new image, whose values $r(x, y)$ encode the presence of a road marking, is then computed according to the following expression:

$$r(x, y) = \begin{cases} d_{+m}(x, y) + d_{-m}(x, y) & \text{if } (d_{+m}(x, y) > 0) \wedge (d_{-m}(x, y) > 0) \\ 0 & \text{otherwise} \end{cases} \quad (14.1)$$

where

$$\begin{cases} d_{+m}(x, y) = b(x, y) - b(x, y + m) \\ d_{-m}(x, y) = b(x, y) - b(x, y - m) \end{cases} \quad (14.2)$$

represent the horizontal brightness gradient. The m parameter is computed according to generic road markings width, image acquisition process, and parameters used in the remapping phase. The resulting filtered image is shown in Figure 14.1(c).

Due to different light conditions (e.g. in presence of shadows), pixels representing road markings may have different brightness, yet maintain their superiority relationship with their horizontal neighbours. Consequently, since a simple threshold seldom gives

a satisfactory binarization, the image is enhanced exploiting its vertical correlation; then an adaptive binarization is performed.

The enhancement of the filtered image is performed through a few iterations of a *geodesic morphological dilation* with the following binary structuring element  , where

$$c(x, y) = \begin{cases} 1, & \text{if } r(x, y) \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{14.3}$$

is the *control image*. The result of the geodesic dilation is the product between the control image and the maximum value computed amongst all the pixels belonging to the neighbourhood described by the structuring element. The iterative application of the geodesic dilation corresponds to the widening of the neighbourhood, except towards the directions in which the control values are 0. Figure 14.2 shows the results of three iterations on a portion of an image that represents a lane marking.

Moreover, since

$$r(x, y) \neq 0 \Longrightarrow \begin{cases} r(x, y - m) = 0 \\ r(x, y + m) = 0, \end{cases} \tag{14.4}$$

the filtered image's pixels at a distance m from a road marking assume a zero value; according to Equation 14.3 their control value is 0 and as a result they form a barrier to the propagation of the maximum value. The enhanced image (after eight iterations) is shown in Figure 14.1(d).

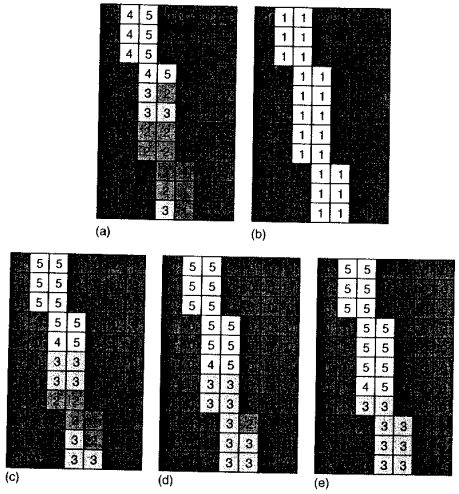


Fig. 14.2 The geodesic morphological dilation used for image enhancement: (*a*) input image, (*b*) control image, and (*c*–*e*) results of the first three iterations.

Finally, the binarization is performed by means of an adaptive threshold:

$$t(x, y) = \begin{cases} 1, & \text{if } e(x, y) \geq \dfrac{m(x, y)}{k} \\ 0, & \text{otherwise} \end{cases}, \tag{14.5}$$

where $e(x, y)$ represents the enhanced image, $m(x, y)$ the maximum value computed in a given $c \times c$ neighbourhood, and k is a constant. The result of the binarization of Figure 14.1(d) (considering $k = 2$ and $c = 7$) is presented in Figure 14.1(e).

Generation of higher-level data structures The binary image is scanned row by row in order to build chains of eight connected non-zero pixels. Following the choice of $m = 2$ at the previous step, every non-zero pixel can have at the most one adjacent non-zero pixel in the horizontal direction; as a result for each pair of horizontally adjacent pixels the one farthest to the right is chosen so that chains are built with a thickness of one pixel.

Each chain is approximated with a *polyline* made of one or more segments, by means of an iterative process. Firstly, the two extrema (A and B) of the polyline are determined averaging the position of the last few pixels at either end of the chain, in order to cope with possible ripples. The segment that joins these two extrema is considered. A first approximate segment ($\overline{AB}$), built using these two extrema, is considered and the horizontal distance between its mid point M and the chain is used to determine the quality of the approximation. In case it is larger than a threshold, two segments ($\overline{AM'}$ and $\overline{M'B}$) sharing an extremum are considered for the approximation of the chain, where M' is the intersection between the chain and the horizontal line that passes through M (see Figure 14.3). The process is iterated on the approximate segments $\overline{AM'}$ and $\overline{M'B}$; at the end of the processing all chains will be approximated by polylines (see Figure 14.1(f)).

High-level processing for lane detection

After the first low-level stage, in which the main features are localized, and after the second stage, in which the main features are extracted, the new data structure (a list of polylines) is now processed in order to semantically group homologous features and to produce a high-level description of the scene.

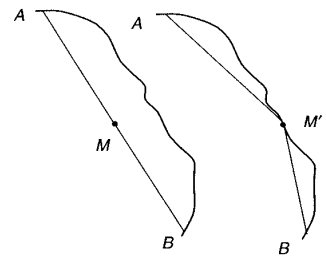


Fig. 14.3 Approximation of chains with polylines.

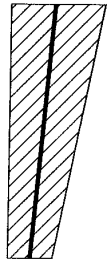


Fig. 14.4 Selection of polylines almost matching the previous left result.

This process in divided into: filtering of noisy features and selection of the feature that most likely belongs to the line marking; joining of different segments in order to fill gaps caused by occlusions, dashed lines, or even worn lines; selection of the best representative and reconstruction of the information that may have been lost, on the basis of continuity constraints; then the result is kept for reference in the next frames and displayed onto the original image.

Feature filtering and selection Each polyline is compared against the result of the previous frame, since continuity constraints provide a strong and robust selection procedure. The distance between the previous result and each extremum of the considered polyline is computed: if all the polyline extrema lay within a stripe centred onto the previous result then the polyline is marked as useful for the following process. This stripe is shaped so that it is small at the bottom of the image (namely close to the vehicle, therefore short movements are allowed) and larger at the top of the image (far from the vehicle, where also curves that appear quickly must be tracked). This process is repeated for both left and right lane markings.

Figure 14.4 shows the previous result with a heavy solid line and the search space with a gridded pattern; it refers to the left lane marking.

Polylines joining Once the polylines have been selected, all the possibilities are checked for their joining. In order to be joined, two polylines must have similar direction; must not be too distant; their projections on the vertical axis must not overlap; the higher polyline in the image must have its starting point within an elliptical portion of the image; in case the gap is large also the direction of the connecting segment is checked for uniform behaviour. Figure 14.5 shows that polyline A cannot be connected to: B due to high difference of orientation; C due to high distance (does not lay within the ellipse); D due to the overlapping of their vertical projections; E since their connecting segment would have a strongly mismatching orientation. It can only be connected to F.

Selection of the best representative All the new polylines, formed by concatenations of the original ones, are then evaluated. Starting from a maximum score, each of the following rules provides a penalty. First each polyline is segmented; in case the polyline does not cover the whole image, a penalty is given. Then, the polyline length

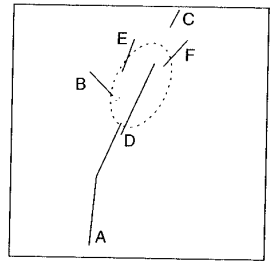


Fig. 14.5 Joining of similar polylines.

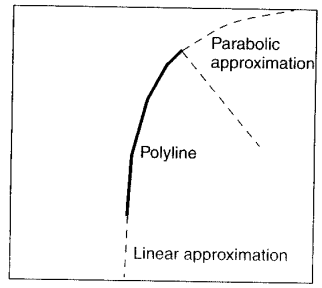


Fig. 14.6 Continuation of short polylines.

is computed and a proportional penalty is given to short ones, as well as to polylines with extremely varying angular coefficients. Finally, the polyline with the highest score is selected as the best representative of the lane marking.

Reconstruction of lost information The polyline that has been selected at the previous step may not be long enough to cover the whole image; therefore a further step is necessary to extend the polyline. In order to take into account road curves, a parabolic model has been selected to be used in the prolongation of the polyline in the area far from the vehicle. In the nearby area, a linear approximation suffices. Figure 14.6 shows the parabolic and linear prolongation.

Model fitting The two reconstructed polylines (one representing the left and one the right lane markings) are now matched against a model that encodes some more knowledge about the absolute and relative positions of both lane markings on a standard road. A model of a pair of parallel lines at a given distance (the assumed lane width) and in a specific position is initialized at the beginning of the process; a specific learning phase allows to adapt the model to errors in camera calibration (lines may be non-perfectly parallel). Furthermore, this model can be slowly changed during the processing to adapt to new road conditions (lane width and lane position), thanks to a learning process running in the background.

The model is kept for reference: the two resulting polylines are fitted to this model and the final result is obtained as follows. First the two polylines are checked for non-parallel behaviour; a small deviation is allowed since it may derive from vehicle movements or deviations from the flat road assumption, that cause the calibration to be temporarily incorrect (diverging of converging lane markings). Then the quality of the two polylines, as computed in the previous steps, is matched: the final result will be attracted toward the polyline with the highest quality with a higher strength.

In this way, polylines with equal or similar quality will equally contribute to the final result; on the other hand, in case one polyline has been heavily reconstructed, or is far from the original model, or is even missing, the other polyline will be used to generate the final result. The weights for the left and right polylines are computed; then, each horizontal line of the two polylines is used to compute the final results (see Figure 14.7).

Finally, Figure 14.8 presents the resulting images referring to the example presented in Figure 14.1. It shows the results of the selection, joining, and matching phases for the left (upper row) and for the right (bottom row) lane markings.

Figure 14.9 presents the final result of the process.

Results of lane detection

This subsection presents a few results of lane detection in different conditions (see Figure 14.10) ranging from ideal situations to road works, patches of non-painted roads, and entries and exits from tunnels. Both highway and extra-urban scenes are provided for comparison; the system proves to be robust with respect to different illumination situations, missing road signs, and overtaking vehicles which occlude the visibility of the left lane marking. If two lines are present – a dashed and a continuous one – the system selects the continuous one.

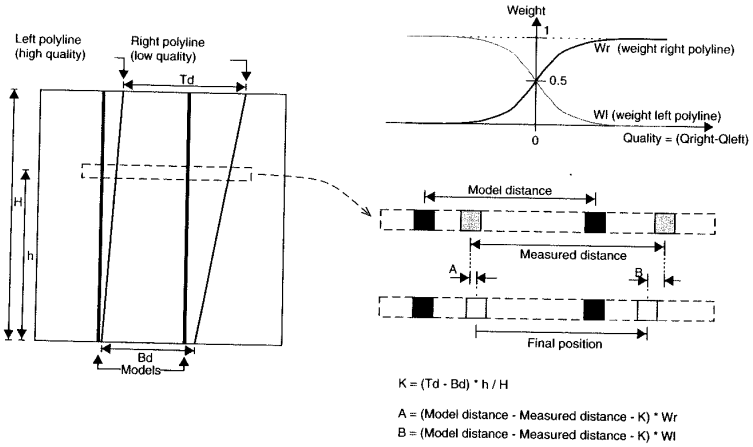


Fig. 14.7 Generation of the final result.

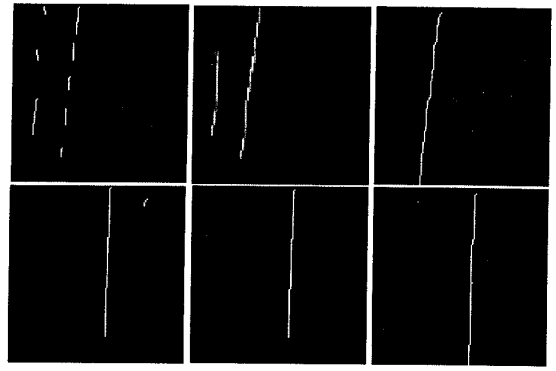


Fig. 14.8 Filtered polylines, joined polylines, and model fitting for the left (upper row) and right (bottom row) lane markings.

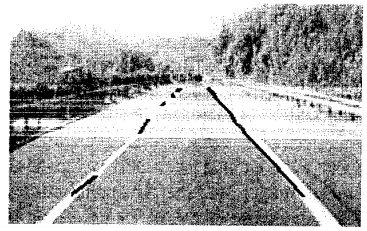


Fig. 14.9 The result of lane detection: black markers represent actual road markings while white markers represent interpolations between them.

14.2.3 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 3D reconstruction is not required and a matching with a given model is sufficient: the model represents the environment without obstacles, and any deviation from the model detects a potential obstacle. In this case the application of IPM to stereo images (Bertozzi *et al.*, 1998b), in conjunction with *a priori* knowledge on the road shape, plays a strategic role.

Low-level processing for obstacle detection

Assuming a *flat road* hypothesis, IPM is performed on both stereo images. 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

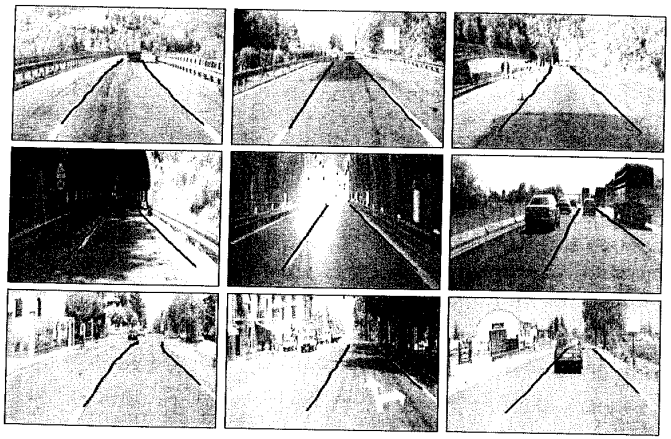


Fig. 14.10 Some results of lane detection in different conditions.

non-zero 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 (Broggi *et al.*, 1999). This behaviour is shown in Figure 14.11, which depicts the left and right views of an ideal white square obstacle on a gridded dark background, the two corresponding remapped images, and the thresholded difference also showing the overlapping between the two viewing areas.

Unfortunately due to the texture, irregular shape, and non-homogeneous 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 quasi-triangular shape are anyway recognizable, even if they are not clearly disjointed. Moreover, if 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, detailed in Figure 14.12, is consequently reduced to the computation of difference between the two remapped images, a threshold, and a morphological opening aimed at removing small-sized details in the thresholded image.

Medium- and high-level processing for obstacle detection

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 (Fascioli, 2000). 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.

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 overthreshold pixels for every straight line originating from the focus. The polar histogram's values are then normalized using the polar histogram obtained by scanning an image where all pixels are set (*reference image*). Furthermore, a low-pass filter is applied in order to decrease the influence of noise (see Figure 14.12(f) and (g)).

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 corresponding to each triangle (Broggi, 1999). 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 (Figure 14.13).

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 (see Figure 14.15).

Peaks joining Two or more peaks can be joined according to different criteria, such as similar amplitude, closeness, or sharpness. The analysis of a large number of different situations made possible the determination of a parameter embedding all of the above quantities. According to the notations of Figure 14.14, R is defined as the ratio between areas A_1 and A_2. If R is greater than a threshold, two adjacent peaks are considered as generated by the same obstacle, and then joined; otherwise, when the two peaks are far apart or the valley is too deep they are left alone (not joined). Figure 14.15 shows some examples of peak joining.

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.

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 α_i of the sector is determined as the width of the polar histogram peak in correspondence to 80 per cent of the peak maximum amplitude h_i. The number of overthreshold pixels is computed and the result is normalized. The radial histogram is analysed 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 (see Figure 14.16).

Results of obstacle detection

Figure 14.17 shows the results obtained in a number of different situations. The result is displayed with black markings superimposed on a brighter version of the left image; they encode both the obstacle's distance and width.

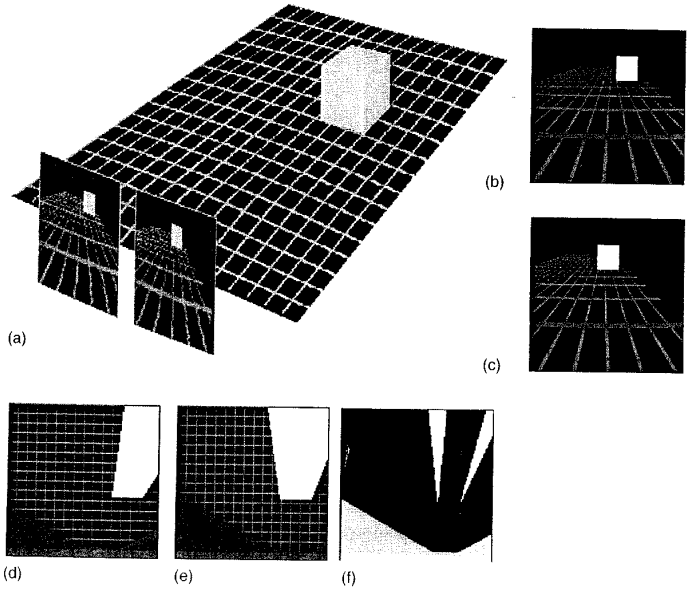


Fig. 14.11 The acquisition of an ideal homogeneous square obstacle: (*a*) 3D representation; (*b*) left image; (*c*) right image; (*d*) left remapped image; (*e*) right remapped image; (*f*) difference image in which the grey area represents the region of the road unseen by both cameras.

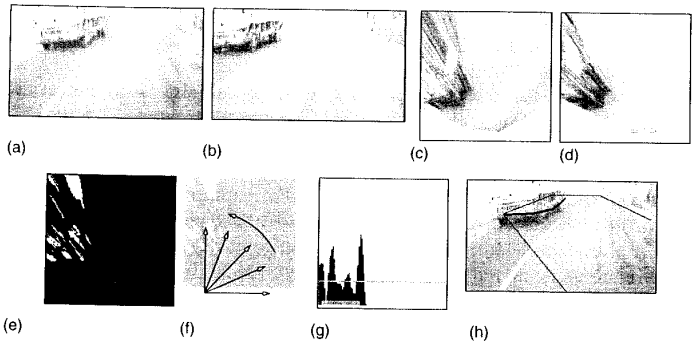


Fig. 14.12 Obstacle detection: (*a*) left and (*b*) right stereo images, (c) and (*d*) the remapped images, (e) the difference image, (*f*) the angles of view overlapped with the difference image, (*g*) the polar histogram, and (*h*) the result of obstacle detection using a black marking superimposed on the acquired left image; the thin black line highlights the road region visible from both cameras.

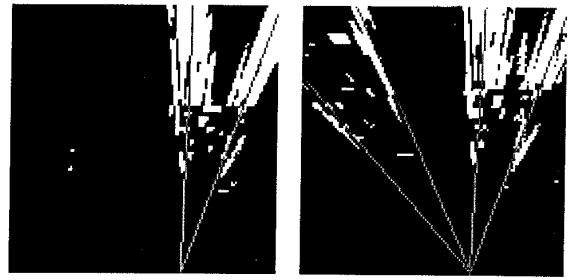


Fig. 14.13 Correspondence between triangles and directions pointed out by peaks detected in the polar histogram.

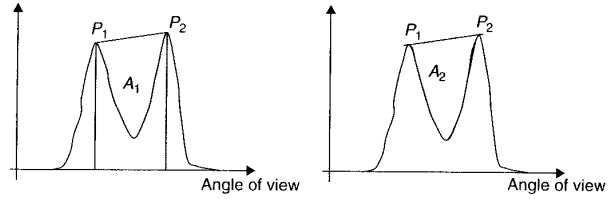


Fig. 14.14 If the ratio between areas A_1 and A_2 is greater than a threshold, the two peaks are joined.

14.2.4 Vehicle detection

The platooning task is based on the detection of the distance, speed, and heading 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 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. These features are used to identify vehicles in the image in the following way: 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 grey level symmetries are considered, but vertical and horizontal edges symmetries as well, in order to increase the detection robustness. Once the symmetry position and width has been detected, a new search begins, which is aimed at the detection of the two bottom corners 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 into the bounding box of the previous frame (partially stretched and reduced to take into account small size variations due to the increment and reduction of the relative distance) and the new frame.

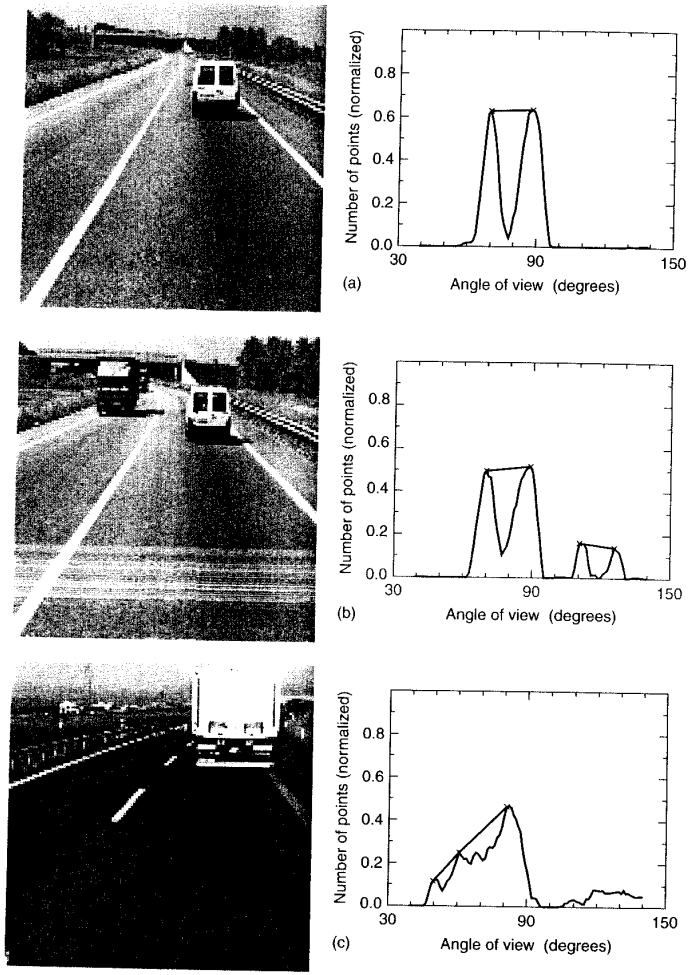


Fig. 14.15 Some examples of peaks join: (*a*) one obstacle, (*b*) two obstacles and (*c*) a large obstacle.

Symmetry detection

In order to search for symmetrical features, the analysis of grey level images is not sufficient. Figure 14.18 shows that strong reflections cause irregularities in vehicle symmetry, while uniform areas and background patterns present highly correlated symmetries. In order to get rid of these problems, also symmetries in other domains are computed.

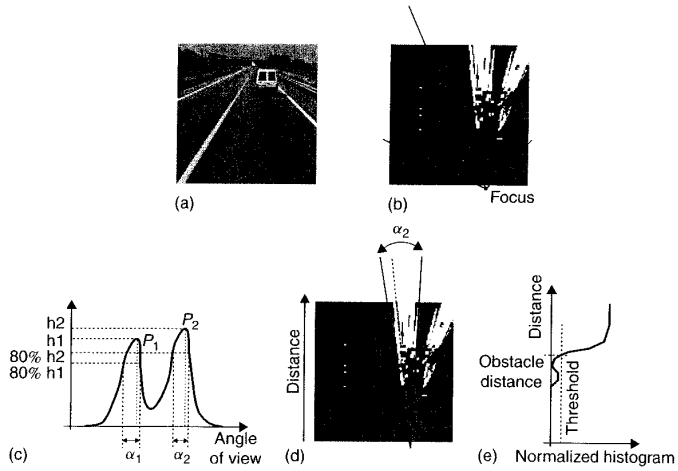


Fig. 14.16 Steps involved during the computation of radial histogram for peak P_2: (*a*) original image; (*b*) binary difference image; (*c*) polar histogram; (*d*) sector used for the computation of the radial histogram; (*e*) radial histogram.

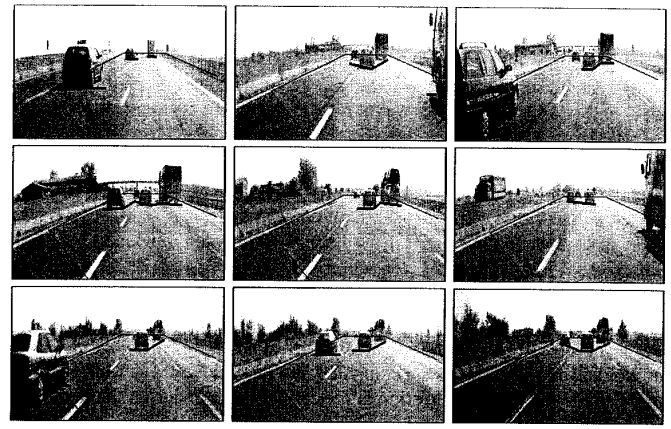


Fig. 14.17 Obstacle detection: the result is shown with a black marking superimposed onto a brighter version of the image captured by the left camera; a black thin line limits the portion of the road seen by both cameras.

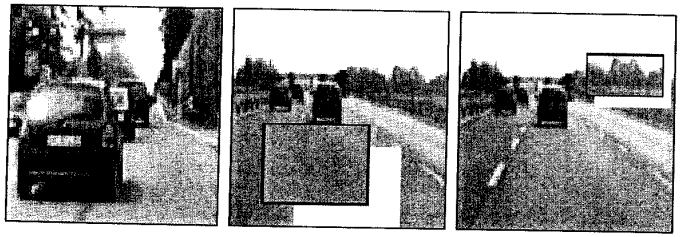
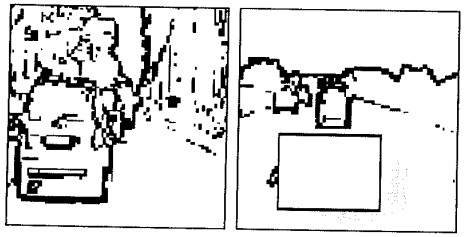


Fig. 14.18 Typical road scenes: in the leftmost image a strong sun reflection reduces the vehicle grey level symmetry; in the centre image a uniform area can be regarded as a highly symmetrical region; the rightmost image shows background symmetrical patterns.



(a) (b)

Fig. 14.19 Edges enforce the detection of real symmetries: strong reflections have lower effects while uniform areas are discarded since they do not present edges.

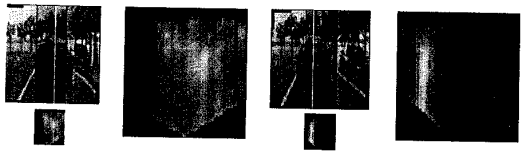


Fig. 14.20 Grey level symmetries: symmetry maps encoding high symmetries with bright points are shown in an enlarged version.

To get rid of reflections and uniform areas, vertical and horizontal edges are extracted and thresholded, and symmetries are computed into these domains as well. Figure 14.19 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. Figure 14.20 shows two examples in which grey level symmetries can be successful for vehicle detection, while Figure 14.21 shows the result of edge symmetry.

For each image, the search area is shown in dark grey 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

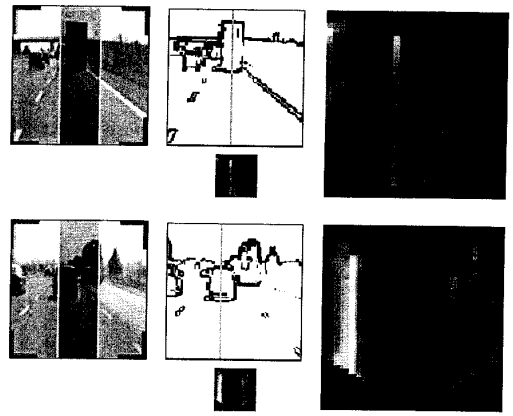


Fig. 14.21 Edge symmetries: the symmetries are computed on the binary images obtained after thresholding the gradient image.

the presence of high symmetries. The 2D symmetry maps are computed by varying the axis' horizontal position within the grey area (shown in the original image) and the symmetry horizontal size. 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 – can be combined with the previous ones to form a single symmetry map. Figure 14.22 shows all symmetry maps and the final one, that allows to detect the vehicle.

Bounding box detection

After the localization of the symmetry, the width of the symmetrical region is checked for the presence of two corners representing the bottom of the bounding box around the vehicle. Perspective constraints as well as size constraints are used to reduce the search. Figure 14.23 shows possible and impossible bottom parts of the bounding box, while Figure 14.24 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 location is again determined by perspective and size constraints. Figure 14.25 shows the search area.

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. Figure 14.26 shows a situation in which the first symmetry maximum, 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.

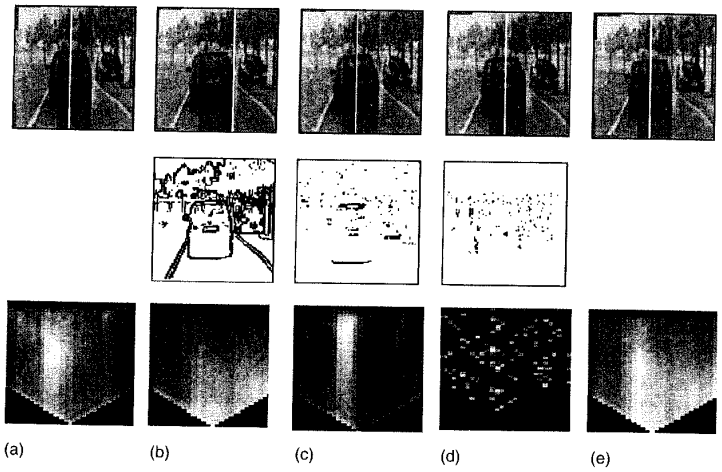


Fig. 14.22 Computing the resulting symmetry: (*a*) grey-level symmetry; (*b*) edge symmetry; (*c*) horizontal edges symmetry; (*d*) vertical edges symmetry; (*e*) total symmetry. For each column the resulting symmetry axis is superimposed onto the top original image.

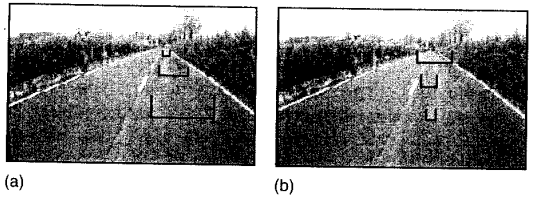


Fig. 14.23 Detection of the lower part of the bounding box: (*a*) correct position and size, taking into consideration perspective constraints and knowledge on the acquisition system setup, as well as typical vehicles' size; (*b*) incorrect bounding boxes.

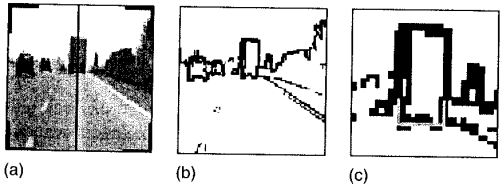


Fig. 14.24 Detection of the lower part of the bounding box: (*a*) original image with superimposed results; (*b*) edges; (*c*) localization of the two lower corners.

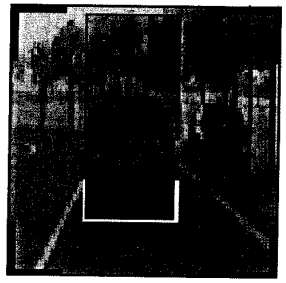


Fig. 14.25 The search area for the upper part of the bounding box is shown in dark grey. It takes into account knowledge about the typical vehicle's aspect ratio.

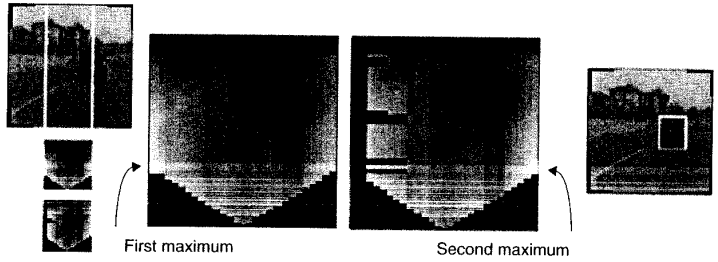


Fig. 14.26 A case in which the total background symmetry is higher than the vehicle symmetry. The search for the bounding box reveals that the left symmetry axis does not correspond to a valid vehicle. The backtracking approach removes the peak near the maximum from the symmetry map and spawns a new bounding box search.

Results of vehicle detection

Figure 14.27 shows some results of vehicle detection in different situations.

14.2.5 Pedestrian detection

A new functionality, still under development and test, allows detection of pedestrians in a way similar to what happens for vehicle detection.

Namely, a pedestrian is defined as a symmetrical object with a specific aspect ratio. With very few modifications to the vehicle detection algorithm presented in the previous section, the system is able to locate and track single pedestrians.

Moreover a new phase, based on stereo vision, aimed at the precise computation of pedestrian distance is currently under study.

Figure 14.28 shows a few examples of correct detection.

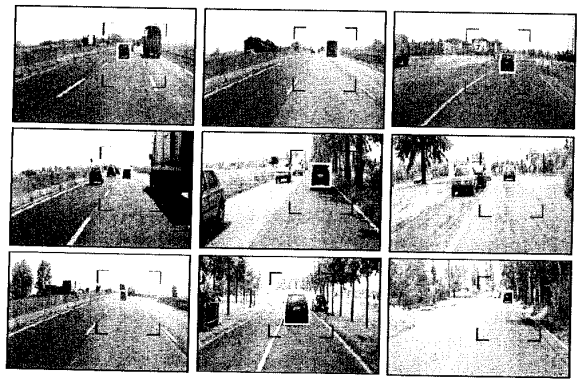


Fig. 14.27 Results of vehicle detection in different road scenes.

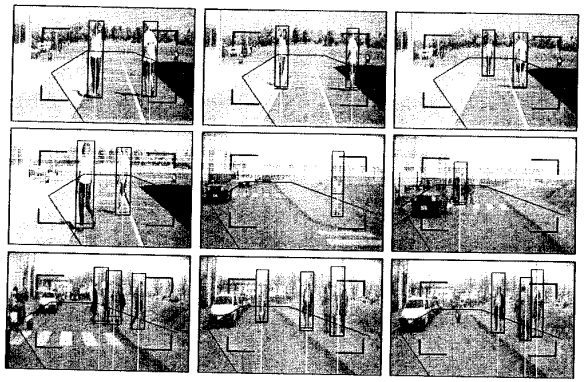


Fig. 14.28 Some preliminary results of pedestrian detection.

14.2.6 The software system's architecture

GOLD is a complex software system made of several interacting modules, all written in C language for a total amount of more than 10 000 code lines.

The program flow chart is sketched in Figure 14.29. Different processing stages can be recognized: after a beginning phase devoted to preliminary actions such as parameter configuration, storage allocation, variables and devices initialization, the main processing cycle is entered and repeatedly executed until termination is requested. Some concluding operations, such as storage deallocation and device closing, are performed before termination.

At each cycle data are first received from different sources: environmental data are acquired from the sensors (cameras and speedometer) into the computer memory, and

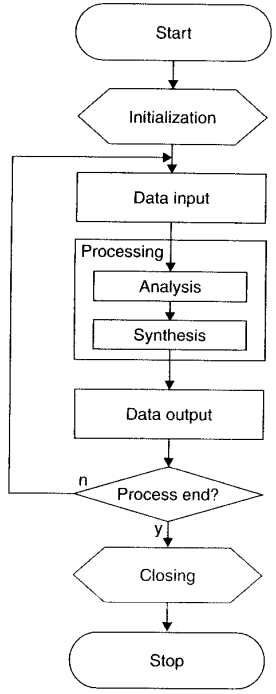


Fig. 14.29 The program flow chart.

commands are received from the user interface. The subsequent processing phase is made of two steps: the analysis of the input data, aimed at the extraction of information concerning the environment, and the exploitation of this knowledge in the decision on the actions to carry on regarding the warning and driving strategies. The data output, therefore, consists in controlling the steering wheel and issuing acoustical and optical warnings. For debugging purposes a visual output can also be supplied to the user and intermediate results can be recorded on disk.

Figure 14.30 sketches the organization and relations among the different program modules from the data flow point of view. As shown, data input and output is performed by means of a modular hierarchical structure: the multiple modules in charge of acquisition or output rely on lower-level modules which interact with the devices' drivers or supply system services. This division between the front end and the back end both in the input and in the output eases the integration of new boards and peripheral devices.

Depending on the configuration parameters, images can be acquired from the cameras, by means of the frame grabber module, when the program is run on board

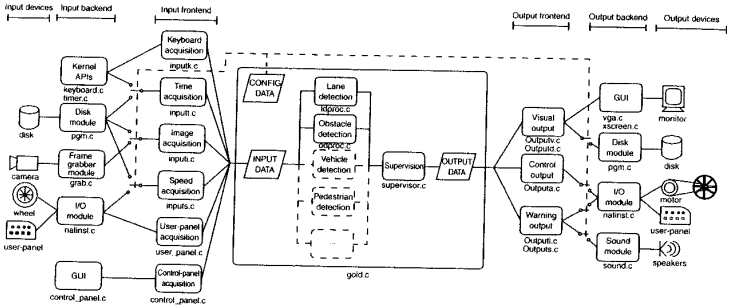


Fig. 14.30 The GOLD system's software architecture. For each module the names of the source files implementing it are displayed.

of the vehicle, or from mass storage (hard or compact disks), through disk managing routines, when it is being tested in the laboratory on pre-recorded image sequences. In the former case the vehicle's speed and the time elapsed between consecutive acquisitions, which are used to assess the spatial distance between frames, are obtained from the speedometer and from the system clock by means of I/O routines and interaction with the kernel APIs, respectively. Conversely, in the latter case they are read from a specific position in the image file where they have been previously encoded during acquisition.

User commands can be entered through the keyboard exploiting the kernel APIs, through the mouse thanks to a software control-panel realized by means of graphical routines, or through a hardware button-based user-panel integrated in the vehicle, managed by the acquisition module through I/O routines.

Several independent modules realize the different functionalities related to the environmental information extraction. The lane detection and obstacle detection functionalities have been initially developed, a module for the localization of the leading vehicle for platooning is currently under development, and, anyway, thanks to the modular structure of GOLD, other modules could be easily integrated, such as the detection of pedestrians or the recognition of road signs.

A supervisor module collects the results from all these processing modules and, by applying strategical reasoning and decisional capabilities, produces the data to output.

Data to control the vehicle trajectory are issued to the motor which drives the steering wheel through I/O modules, which are also used as an interface towards the warning LEDS integrated in the on-board user panel.

The visual output for debugging is produced by means of the routines of the graphical user interface and, depending on the available graphic environment, can be displayed under the X windowing system or can be directly sent to the VGA.

The full modularity and hierarchical structure of the GOLD software architecture allows expanding the system by adding new functionalities, guarantees portability, hardware independence and reconfigurability by allowing the use of different devices, and eases the debugging and testing of the algorithms.

14.2.7 Computational performance

Table 14.1 shows the timing performance obtained on the computing system currently installed on ARGO (see Section 14.3.3); since obstacle and lane detection 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.

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 Figure 14.31).

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

Table 14.1 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

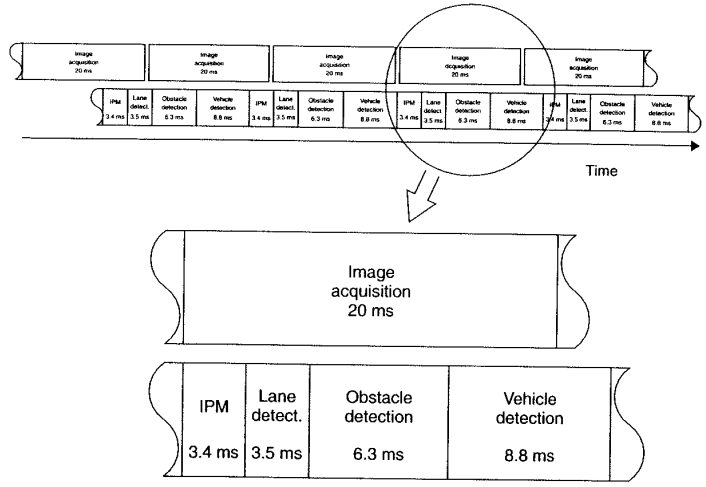


Fig. 14.31 Performance of image acquisition and processing.

14.3 The ARGO prototype vehicle

ARGO, shown in Figure 14.32, is an experimental autonomous vehicle equipped with vision systems and an automatic steering capability.

It is able to determine its position with respect to the lane, to compute the road geometry, to detect generic obstacles on the path, and to localize a leading vehicle. The images acquired by a stereo rig placed inside the windscreen are analysed in real time by a computing system located into 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 behaviour 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 automatic driving capabilities fully.

14.3.1 Functionalities

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

- **Manual driving:** the system simply monitors and logs the driver's activity.
- **Supervised driving:** in case of danger, the system warns the driver with acoustic and optical signals.

Fig. 14.32 The ARGO prototype vehicle.

- **Automatic driving:** the system maintains the full control of the vehicle's trajectory, and the two following functionalities can be selected:
  - **Road following:** consisting of 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:** namely the automatic following of the preceding vehicle, that requires the localization and tracking of a target vehicle (*vehicle detection and tracking*), and relies on the recognition of specific vehicle's characteristics.

14.3.2 The data acquisition system

Only passive sensors (two cameras and a speedometer) are used on ARGO to sense the surrounding environment: although vision is extremely computationally demanding, it offers the possibility to acquire data in a non-invasive way, namely without altering and polluting the environment. 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.

The vision system

The ARGO vehicle is equipped with a stereoscopic vision system consisting of two synchronized cameras able to acquire pairs of grey level images simultaneously. The installed devices are small (3.2 cm $\times$ 3.2 cm) low-cost cameras featuring a 6.0 mm focal length and 360 lines resolution, and can receive the synchronism from an external signal.

The cameras lie inside the vehicle at the top corners of the windscreen, so that the longitudinal distance between the two cameras is maximum. This allows the detection of the third dimension at long distances. The optical axes are parallel and, in order to also handle non-flat roads, a small part of the scene over the horizon is also captured, even if the framing of a portion of the sky can be critical due to abrupt changes in image brightness: in case of high contrast, the sensor may acquire over-saturated images.

The images are acquired by a PCI board, which is able to grab three 768 $\times$ 576 pixel images simultaneously. The images are directly stored into the main memory of the host computer thanks to the use of DMA. The acquisition can be performed in real time, at 25 Hz when using full frames or at 50 Hz in the case of single field acquisition.

System calibration

Since the process is based on stereo vision, camera calibration plays a fundamental role in the success of the approach. The calibration process is divided into two steps:

- **Supervised calibration:** the first part of the calibration process is an interactive step: a grid with a known size, shown in Figure 14.33, has been painted onto the ground and two stereo images are captured and used for the calibration. Thanks to an X Windows-based graphical interface a user selects the intersections of the grid lines using a mouse; these intersections represent a small set of points whose world coordinates are known to the system; this mapping is used to compute the

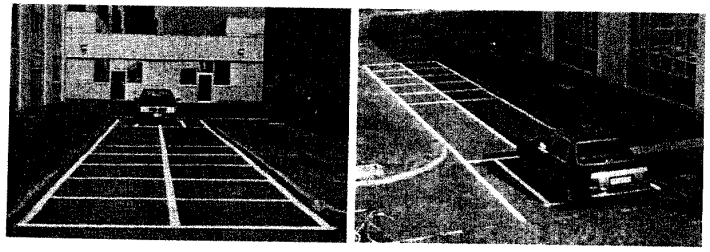


Fig. 14.33 View of the calibration grid painted on a reference road segment.

calibration parameters. This set of homologous points is used to minimize different cost functions, such as the distance between each point and its neighbours as well as line parallelism.

This first step is intended to be performed only once when the orientation of the cameras or the vehicle trim has changed. Since the homologous points are few and their coordinates may be affected by human imprecision, this calibration represents only a rough guess of the parameters, and a further process is required.

- **Automatic parameters tuning:** after the supervised phase, the computed calibration parameters have to be refined. Moreover, small changes in the vision system setup or in the vehicle trim require a periodic tuning of the calibration. For this purpose, an automatic procedure has been developed (Bertozzi *et al.*, 1998b). Since this step is only a refinement, a structured environment, such as the grid, is no longer required and a mere flat road in front of the vision system is sufficient. The parameters' tuning consists of an iterative procedure, based on the application of the IPM transform to stereo images, which takes about 20 seconds: iteratively small deviations from the coarse parameters computed during the previous step are used to remap the captured images; the aim is to get the remapped images as similar as possible. All the pixels of the remapped images are used to test the correctness of the calibration parameters through a least square difference approach.

The speed sensor

The vehicle is also equipped with a speedometer to detect its velocity. A Hall effect-based device has been chosen due to its simplicity and its reliability and has been interfaced to the computing system via a digital I/O board with event counting facilities. The resolution of the measuring system is about 9 cm/s.

The user interface

Finally, a set of buttons on a control panel allows the user to interact with the system: the driver can select the functionality, modify some driving parameters, and – in this first stage – can use it as an interface for a basic debugging tool. The control panel is shown in Figure 14.34.

The keyboard

The on-board PC's keyboard is used as a set of auxiliary buttons to trigger on or off specific debugging facilities and modify the internal driving parameters that are hidden from the driver. It is therefore only used to debug and tune the system's parameters.

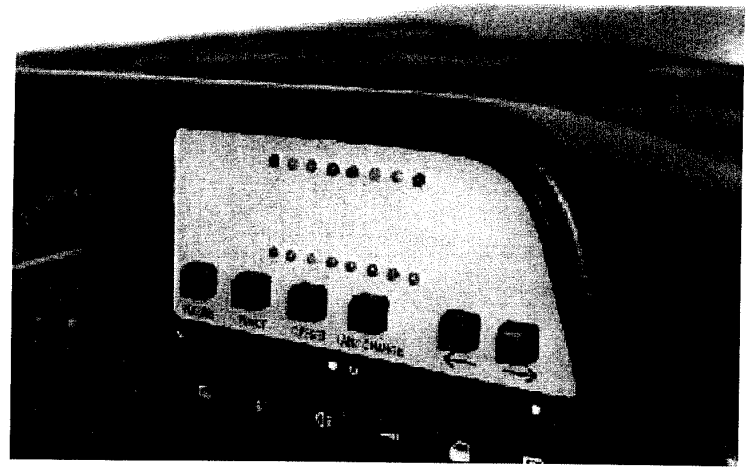


Fig. 14.34 The control panel, which includes the two sets of LEDs and a series of buttons.

14.3.3 The processing system

Two different architectural solutions were considered and evaluated: a special-purpose and a general-purpose processing system. Although the dedicated architecture presents the advantages of an *ad hoc* solution, it requires an expensive and complex design, whilst, on the other hand, the standard architectural solutions offer nowadays a sufficient computational power even for hard real-time constrained applications as this one.

For these reasons, the architectural solution which was originally installed and tested on the ARGO vehicle was based on a standard 200 MHz MMX Pentium processor. Software performance was boosted exploiting the SIMD MMX enhancements of the traditional instruction set, which permits processing in parallel multiple data elements using a single instruction flow. The processing system currently installed on ARGO is based on a standard 450 MHz Pentium II processor. Figure 14.35 shows the ARGO boot in which the PC is visible.

14.3.4 The output system

Four different output devices have been installed on ARGO: some of them are actuators, some were integrated in order to test their usefulness, and a few were included just for debugging purposes. Figure 14.40 on shows an internal view of the driving cabin.

The acoustical devices

A pair of stereo loudspeakers are used to issue warnings to the driver in case dangerous conditions are detected, e.g. when the distance from the leading vehicle is under a safety threshold or when the vehicle position within the lane is unsafe. The possibility of using

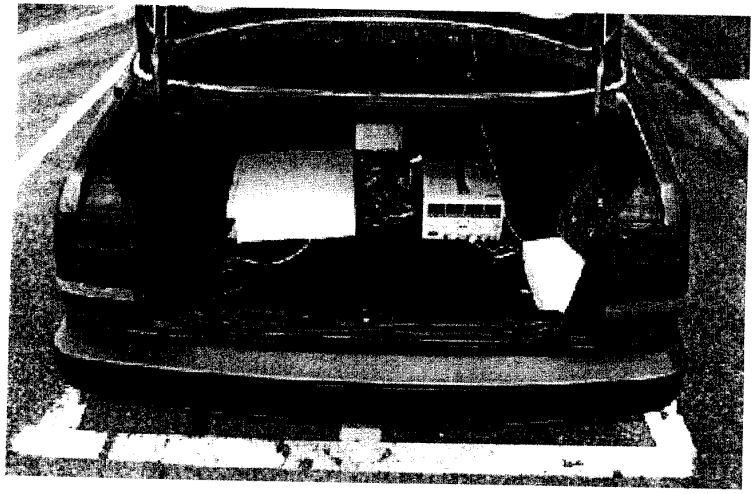


Fig. 14.35 Internal view of ARGO's boot: both the PC and inverter are visible.

the loudspeakers individually is exploited in order to enhance the effectiveness of the acoustical messages. The driver can choose amongst several kinds of acoustic signals: chimes only, vocal messages in different languages (Italian or English) and with a different information content.

The optical devices

Two different optical devices, both shown in Figure 14.36, have been installed: a LED-based control panel and a colour monitor. The former is used to display the current functionality (bottom row LEDs) and a visual information about the relative position of the vehicle compared to the lane, namely the vehicle offset with respect to the road centre line (top row LEDs).

The latter is mainly used as a debugging tool, since its integration on a commercial vehicle would distract the driver's attention: a visual feedback is supplied to the driver by displaying the results of the process on a 6-inch on-board colour monitor (see Figure 14.37). The monitor presents the acquired left image with markings highlighting lane markings as well as the position of obstacles and preceding vehicles. It also presents on the right side a bi-dimensional reconstruction of the road scene, showing the position of obstacles, vehicles and lane markings, and, on the bottom, the vehicle's speed, a software version of the control panel's LEDs, and a measure of the vehicle offset with respect to the road centre line. As a different option it can display images representing intermediate processing results and the value of some basic parameters for debugging purposes.

Figure 14.38 shows the debugging tool, running under the X11 windowing system; it allows tuning of the GOLD parameters and computing of processing statistics.

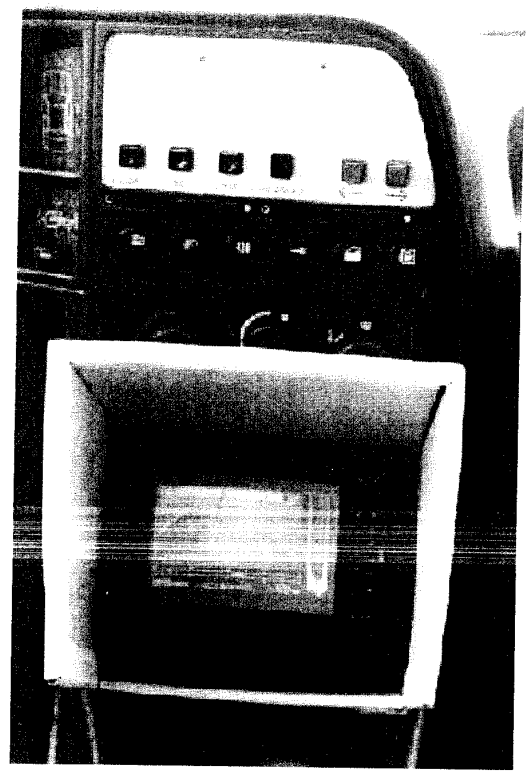


Fig. 14.36 The control panel and on-board monitor.

The mechanical devices

A single mechanical device, shown in Figure 14.39, has been integrated on ARGO to provide autonomous steering capabilities. It is composed of an electric engine linked to the steering column by means of a belt.

This device can operate in three different ways:

- as a *warning* device, it is used to vibrate the steering wheel for a short while in order to warn the driver when an unsafe manoeuvre is undertaken or in correspondence to a momentary reduction in driving capabilities (e.g. drowsiness, wrong security distance, etc.);
- as an *actuating* device, it is used during humanly operated vehicle driving to restore a safe lateral position of the vehicle in case of dangerous situations: it steers the

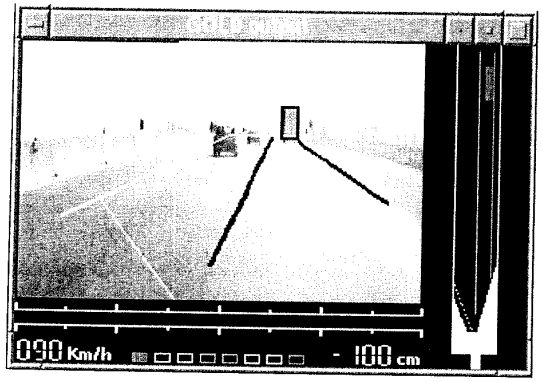


Fig. 14.37 The output of the GOLD system on the on-board monitor.

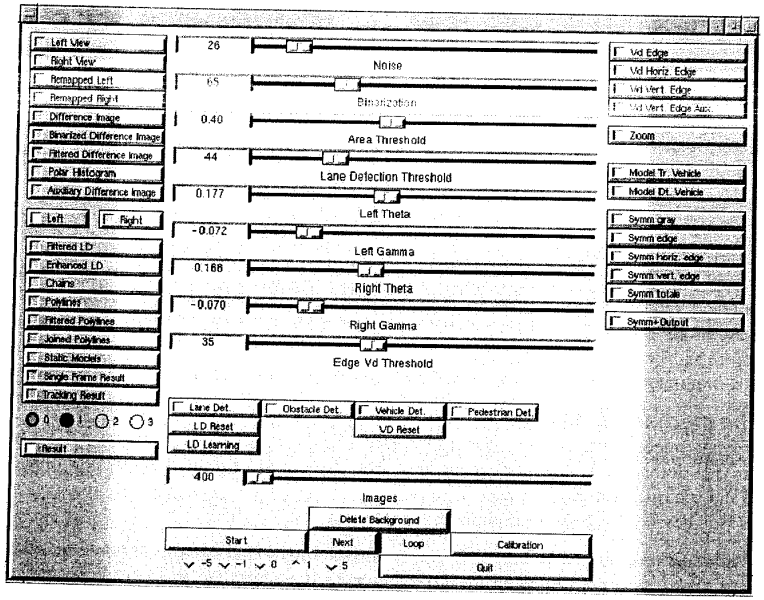


Fig. 14.38 The software control panel used for debugging.

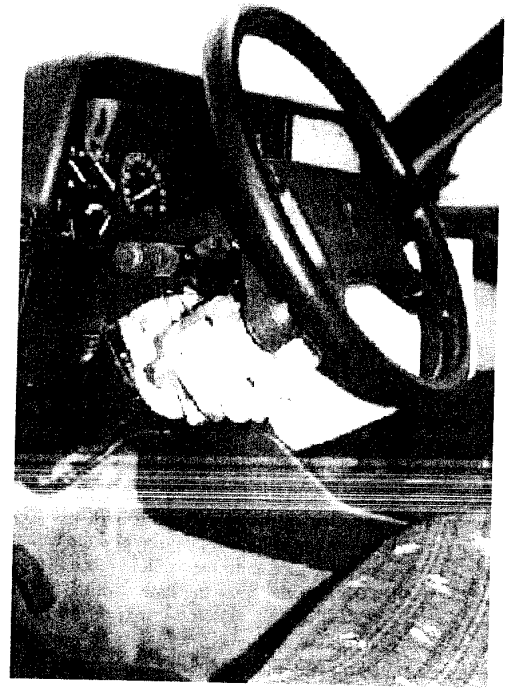


Fig. 14.39 View of the electric engine installed on the steering column.

vehicle and releases the control of the steering wheel to the driver only when the vehicle is in a safe position within the lane;

- again as an *actuating* device, to drive the vehicle autonomously: the output provided by the GOLD vision system, i.e. the lane centre at a given distance in front of the vehicle for road following or the position of the leading vehicle for platooning, are used to steer the wheel so to direct the vehicle towards that point.

14.3.5 The control system

The controller currently adopted for the ARGO vehicle was initially designed and optimized for the road following task. Minor changes have been introduced to implement also the platooning functionality.

Once the road geometry ahead of the vehicle has been recovered, a software module is used to compute signals to be issued to the steering wheel actuator. More precisely, starting from the knowledge regarding road geometry and vehicle speed, this module determines the vehicle yaw and computes the position that the vehicle will assume in a given time interval (in the current implementation it is 1.5 s). In case the future vehicle position is not compatible with the requirements (the vehicle should keep central with respect to the driving lane), a rotation is imposed on the steering wheel.

To keep the control system simple, the steering wheel rotation is computed as the result of a temporal average amongst a set of values proportional to the offsets between the future position and the lane centre in successive time instants. Despite the delay caused by temporal average, this low-pass filter has the specific advantage of reducing noise caused by possible incorrect results of the vision system, due – for example – to calibration drifts induced by vehicle movements.

Since the processing rate is high (25 frames per second or more), the refinements of the steering wheel positions occur at a high rate; this eases the filtering of incorrect, vibrating movements caused by noise.

Moreover, an emergency feature has been added to the control system to avoid sudden and therefore dangerous movements of the steering wheel. Each new position – computed as mentioned above – is compared to the current position and processed through a limiting filter, whose threshold depends on vehicle speed. In other words, when the vehicle is moving at a low speed, two consecutive positions of the steering wheel can be quite different; on the other hand, when driving at a high speed, only small steering wheel movements are permitted.

When the road following functionality is active, by means of specific buttons in the control panel, left or right lane change manoeuvres can be triggered off. In order to change the lane, the offset between the vehicle's position and the lane marking that the system is tracking is progressively varied until the vehicle reaches the central position in the new lane. This implies the knowledge of the lane width. Then the system locks onto the new lane marking (delimiting the new lane), which is easily localized lying parallel to the previous one at a known distance.

The control strategy adopted for platooning takes advantage of the previously defined control scheme. The main difference with respect to the path-following functionality is on the estimation of the offset error. When the platooning functionality is activated, the target point is centred on the preceding vehicle so that the target look-ahead distance is neither constant nor the most appropriate for the current velocity. Obviously, the use of this look-ahead distance and the corresponding offset error could degrade the performance of the functionality. The efficiency of the platooning control algorithm is recovered by scaling the tracking error measured at that distance to an estimated offset error given by a *virtual* target point placed at the appropriate look-ahead distance.

14.3.6 Other vehicle equipments and emergency features

Besides the main devices for input, output and data processing, some additional equipments have been installed on the vehicle.

- Since all internal instrumentation works at 220 V @ 50 Hz, an inverter has been included to provide the correct power supply.

- The vehicle has been connected to the Internet thanks to two GSM cellular channels. The connection is set up on two serial links with the PPP (Point to Point) protocol and the packets are switched onto the two lines for load balancing reasons. This connection permits the transfer of images and statistical data from the vehicle to Internet.
- Finally, the ARGO vehicle has been equipped with several emergency devices to be activated by hand in case of system failures. The availability of these devices has been necessary to ensure safety during the tests and to face insurance requirements.

Besides acting on the control panel to deactivate automatic driving features, the passenger can intervene at different levels on a number of emergency devices (see Figures 14.40 and 14.41).

- **Emergency joystick:** a joystick is used to overcome the commands issued by the system. The passenger, after pushing one of the two joystick buttons, can take control of the steering wheel by moving the joystick. This emergency feature is intended only to refine the commands issued by the system. When the button is released, the system keeps working automatically.
- **Emergency pedal:** located near the clutch, it is used to set the *manual driving* functionality. As soon as the pedal is pressed, the control software switches to manual driving and the passenger can take control of the vehicle.
- **Emergency button:** an emergency button, located near the manual brake, performs via hardware the same action of the emergency pedal, detaching the electric engine control signal.
- **Power supply switch:** this last device can be used in case of the hardware or software system's failure, and allows the release of the power supply for the steering wheel electric engine.

14.4 The MilleMiglia in Automatico test

14.4.1 Description

In order to extensively test the vehicle under different traffic situations, road environments, and weather conditions, a 2000 km journey was carried out in June 1998. Other prototypes were tested on public roads with long journeys (CMU's Navlab *No Hands Across America*, and a tour from Munich to Odense organized by the Universität der Bundeswehr, Germany) whose main differences were that the former was relying also on non-visual information (therefore handling occlusions in a different way) and that the latter was equipped with complex computing engines.

The *MilleMiglia in Automatico* test was carried out in 1998, and the system was much more primitive than it is currently. Only lane detection and obstacle detection were tested: lane detection was based on the localization of a single line, while the detection of the preceding vehicle was performed by the obstacle detection module; no tracking was done and only the road following functionality was available.

In the following subsections the test is described, and a critical discussion is presented. The main bottlenecks of the system, as well as some considerations which led to the current implementation (described in the previous sections) are highlighted.

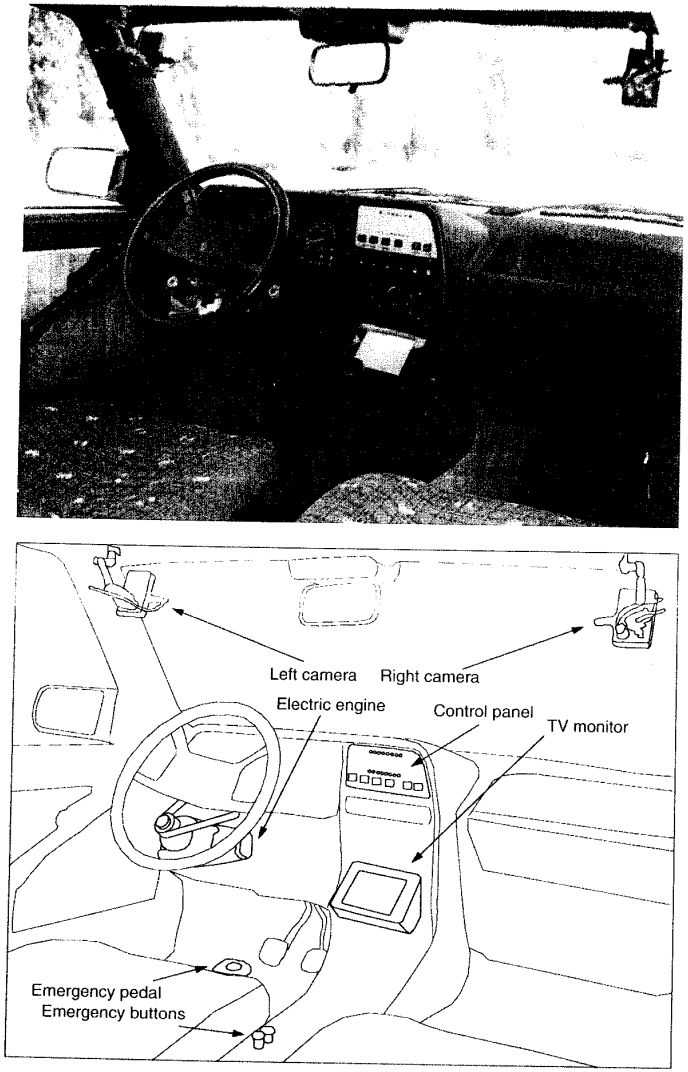


Fig. 14.40 Internal view of the ARGO vehicle.

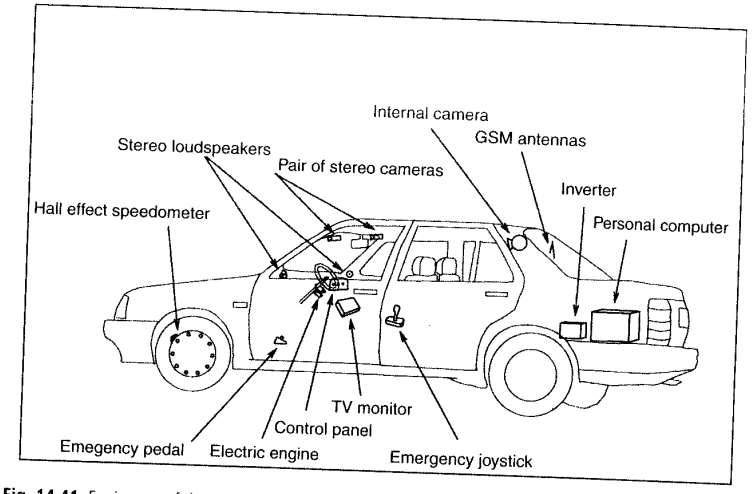


Fig. 14.41 Equipment of the ARGO vehicle.

Test schedule

During this test, ARGO drove itself autonomously along the Italian highway network, passing through flat areas and hilly regions including viaducts and tunnels. The Italian road network is particularly suited for such an extensive test since it is characterized by quickly varying road scenarios which included changing weather conditions and a

Fig. 14.42 Automatic driving during the *MilleMiglia in Automatico* tour.

generally good amount of traffic. The tour took place on highways and freeways, but the system proved to work also on sufficiently structured rural extra-urban roads with no intersections.

The *MilleMiglia in Automatico* tour took place between 1 and 6 June, 1998: it was subdivided into seven stages of about 250 km each, and carried out within six days.

- Monday, 1 June 1998
  9.00–12.00 Parma–Turin ..... 245 km
  Mainly flat.
- Tuesday, 2 June 1998
  9.00–11.00 Turin–Milan–Pavia ..... 175 km
  Flat; the Milan by-pass taken.
  16.00–20.00 Pavia–Milan–Ferrara ..... 340 km
  Flat; again through the Milan by-pass.
- Wednesday, 3 June 1998
  12.00–15.30 Ferrara–Bologna–Ancona ..... 260 km
  The Bologna by-pass taken; flat until Rimini, then hilly crossing the Appennine region towards Ancona.
- Tuesday, 4 June 1998
  8.30–13.00 Ancona–Pescara–Rome ..... 365 km
  Hilly until Rome.
- Friday, 5 June 1998
  9.00–12.30 Rome–Florence ..... 280 km
  Flat and hilly. The GRA (*Grande Raccordo Anulare*) taken.

- Saturday, 6 June 1998
  8.30–12.00 Florence–Bologna–Parma ..... 195 km
  Mountainous: crossing the Appennine region as far as Bologna, then flat until Parma.

Data logging

During the journey, besides the normal tasks of data acquisition and processing for automatic driving, the system logged the most significant data, such as speed, position of the steering wheel, lane changes, user interventions and commands, and dumped the whole status of the system (images included) in correspondence to situations in which the system had difficulties in detecting reliably the road lane.

This data has been processed off-line after the end of the tour in order to compute the overall system performance, such as the percentage of automatic driving, and to analyse unexpected situations. The possibility of re-processing the same images, starting from a given system status, allows the reproduction of conditions in which the fault was detected and a way of solving it is found. At the end of the tour, the system log contained more than 1200 Mbyte of raw data.

Live broadcasting of the event via the Internet

During the tour, the ARGO vehicle broadcasted a live video stream on the Internet: two GSM cellular modems were connected to the Vision Laboratory of the Dipartimento di Ingegneria dell'Informazione in Parma, and used to transfer both up-to-date news on the test's progression and images acquired by a camera installed in the driving cabin, shown in Figure 14.43, demonstrating automatic driving.

The set-up of a permanent data link from a moving platform implies the use of mobile telecommunications facilities, such as GSM modems. The use of GSM modems for live video streaming faces the following constraints:

- low bit-rate for transmission (usually 9600 bps) and
- high bandwidth variability during movements.

In order to increase the throughput of the link, two modems – and therefore two channels – working simultaneously have been installed on ARGO. The transmission hardware is composed of a PC running Linux with two serial ports and a colour camera connected to the parallel port. The network traffic is split across the two serial channels thanks to the EQL (EQualizer Load-balancer) protocol, able to split the network traffic across multiple links. Moreover, the communication software was designed in order to dynamically adapt the network traffic to the throughput.

To prove the scientific community, mass media and general public's high interest, the web site received more than 350 000 contacts during the tour and more than 3000 Mbyte of information were transferred, with a peak of 16 000 contacts per hour during the first day of the tour.

14.4.2 System performance

This section deals with various issues related to system performance. It covers the following: hardware issues regarding the vision system, considerations made on the image processing algorithms and processing speed, issues related to the vehicle control

Fig. 14.43 The internal camera framing the driving cabin; a GSM antenna is also visible in the background.

system and in particular to the visual feedback used in the vehicle control loop, the experience regarding the use of the man-machine interface installed on ARGO, and a discussion on how environmental conditions influence the whole system. This chapter ends with an overview of some of the problems encountered during the tour and a detailed analysis of one hour of automatic driving.

Vision system

Since one of the goals of the whole project is the development of a sufficiently low-cost system so as to ease its integration in a large number of vehicles, the use of low-cost acquisition devices was a clear starting point: in particular videophone cameras (small sized 3.5 cm × 3.5 cm sensors at an average cost of US$100 each) were installed.

Although having a high sensitivity even in low light conditions (such as during the night), a quick change in the illumination of the scene causes a degradation in the image quality (for example at the entrance or exit from a tunnel). In particular, having

a slow automatic gain control (they have been designed for applications characterized by constant illumination such as video-telephony), in correspondence to the exit from a tunnel for a period of time of about $100 \div 200$ ms, the acquired images become completely saturated and their analysis is impossible. This problem also happens when the vehicle crosses an area that contains patches of new (black) and old (light grey or even white) asphalt. A possible solution to this problem could be to use cameras with a higher performance rate and, in particular, a faster automatic gain control and higher dynamics. CMOS-based sensors are also now being evaluated: their logarithmic response should allow higher dynamics, and their intrinsic slowness due to the specific pixel-by-pixel addressing mode, should pose no additional problems since the early stages of processing (IPM) are based on a strong subsampling of the incoming image. Moreover, their use should speed-up the acquisition process since a complete scanning and transfer of the whole image is no longer required.

Processing system

The processing system proved to be powerful enough for the automatic driving of the vehicle. Moreover, current technology provides processing systems with characteristics that are definitely more powerful than the one installed on ARGO during the test: a commercial PC with a 200 MHz Pentium processor and 32 megabytes of memory.

On such a system, enhanced by a video frame grabber able to acquire simultaneously two stereo images (with 768 × 576 pixel resolution), the GOLD system processes up to 25 pairs of stereo frames per second and provides the control signals for autonomous steering every 40 ms.[1] Obviously the processing speed influences the maximum vehicle speed: the higher the processing speed, the higher the maximum vehicle speed.

Visual processing

The approach used for both obstacle and lane detection, based on the inverse perspective mapping transform, proved to be effective for the whole trip. Even if on Italian highways the flat road assumption (required by the IPM transform) is not always valid, apart from exceptions, the approximation of the road surface with a planar surface was acceptable. The calibration of the vision system, in fact, should be tuned to reflect the modifications of road slope ahead of the vehicle, since a wrong calibration generates a lateral offset in the computation of the vehicle trajectory. Nevertheless, since the highway lanes' width is sufficiently large, this offset has never caused serious problems, but it is for this reason that an enhancement to the IPM transform is currently under development allowing to get rid of the flat-road assumption and to also handle generic roads (Bertozzi, 1998a).

The only drawback which exists, due to the use of the IPM transform, is that the vehicle movements (pitch and roll) do not allow a reliable detection of obstacles at distances further than 50 metres.

During navigation the system locked onto the right lane marking (as already mentioned, lane detection was based on the localization of a single line), hence overtaking vehicles did not occlude its visibility. This choice, however, was critical during the complex situations of highway exits, where two lines are present, a continuous one for the exit and a dashed one for the lane. In such cases the user

[1] This is equivalent to one refinement on the steering wheel position for every metre when the vehicle drives at 100 km/h.

could select whether to stay on the road or to exit it. Moreover, occlusions by other vehicles, even if much less frequent as they could have been for the left lane marking, could represent a problem. For this reason a more robust version of the lane detection algorithm (discussed in Section 14.2.2) has been developed, which can handle both lines. This choice enforces the system's reliability also in case either of the two lines is worn or missing.

On the contrary, the resolution of these devices turned out to be satisfactory and the framing of the scene (no horizon to reduce strong light conditions and direct sunlight) correct for the type of roads considered (sloping gently).

During the whole tour the system processed about 1 500 000 images[2] totalling about 330 Gbyte of input data.

Control system

The control system has been designed and developed focusing on its simplicity. Regarding the *mechanical* part, an electric stepping motor allows the rotation of the steering wheel with a high resolution and a reduced power consumption.

On the other hand, the simplicity of the *logical* part of the control system has the main advantage of keeping the entire system robust: only a temporal average and a simple measurement are used to compute the steering wheel angles.

With this kind of control, for speeds reaching around $90 \div 95$ km/h there is no noticeable difference in comparison to a human driver, while for higher speeds (during the test a peak of 123 km/h was reached) the vehicle tends to demonstrate slightly unstable behaviour, oscillating inside the lane.

A more sophisticated control system is currently under study also including a strong road model; this should allow reaching higher speeds with more stability. This new control mechanism is undergoing tests on a system simulator that permits the emulation of vehicle behaviour on curved roads with more than one lane. It has also been used to test lane change manoeuvre.

Man–machine interface

The automatic driving system is managed by a control panel by means of six buttons and eight LEDs which indicate the currently selected driving functionality. Since this is still in an early development stage, it is possible to also modify the value of a number of driving parameters, which should normally be hardcoded and not adjustable by the final user. Nevertheless, this possibility does not increase the interface complexity. For security reasons the user is required to press two buttons simultaneously for each command to be executed. The modifications in the system's state are then notified to the user through vocal messages which confirm the reception and execution of the command. Moreover, for some commands whose execution is not instantaneous (such as the lane change command) an acoustic signal is also generated upon completion of the manoeuvre.

The emergency systems allows the driver to take the control of the vehicle during emergency situations in three different ways: tuning driving parameters (through the control panel buttons), temporarily taking over the system (by means of a joystick), completely taking over the system (by pressing a pedal close to the clutch, or the control panel buttons, or an emergency button close to the handbrake).

[2] Each with size 768 × 288 pixel.

Environmental conditions

During the six-day test, the system's behaviour was evaluated in various environmental conditions. The route was chosen in order to include areas with different morphological characteristics: from flat areas (Parma, Piacenza, Turin, Milan, Verona, Padova, Ferrara, Bologna), to sloping territories of the Appennines region (Pescara, L'Aquila, Rome, Florence, Bologna), and heavy traffic zones (Rome, Turin and Milan's by-passes), inevitably encountering stretches of highway with road works, absent or worn horizontal road signs, and diversions.

Moreover, the different weather conditions (in particular the light conditions) demonstrated the robustness of the image processing algorithms. In fact, the system was always able to extract the information for the navigation task even in critical light conditions, with the sun in front of the cameras, high or low on the horizon, during the evening as well as during the day, with high or low contrast.

The second leg of the tour ended in the late evening; at night-time the system's behaviour is improved by the absence of sunlight reflections and shadows, whilst the area of interest is constantly illuminated by the vehicle headlights.

One of the problems, which is now being solved through shadowing devices and a light polarizing filter, is the light's reflection within the internal surface of the windscreen, as shown in Figure 14.44. Figure 14.45 shows some of the typical effects of this reflection.

Finally, the system proved to be surprisingly robust despite the high temperatures measured during the tour: in some cases the external temperature reached 35°C and the system continued to work reliably even with no air conditioning.

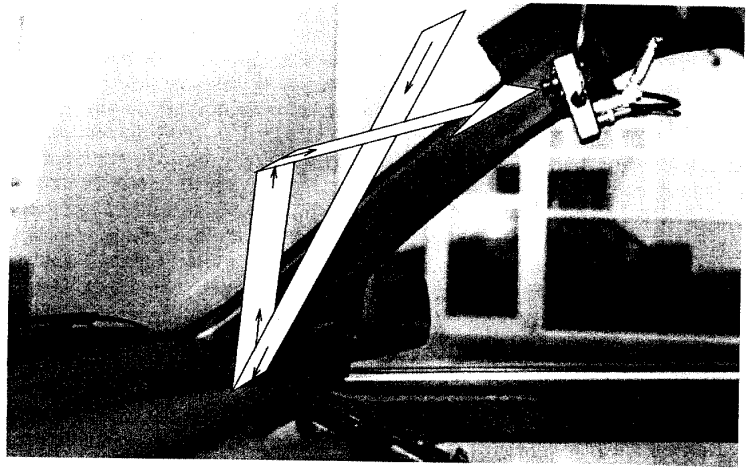


Fig. 14.44 The light's reflection within the internal surface of the windscreen; this causes the images to be over-saturated.

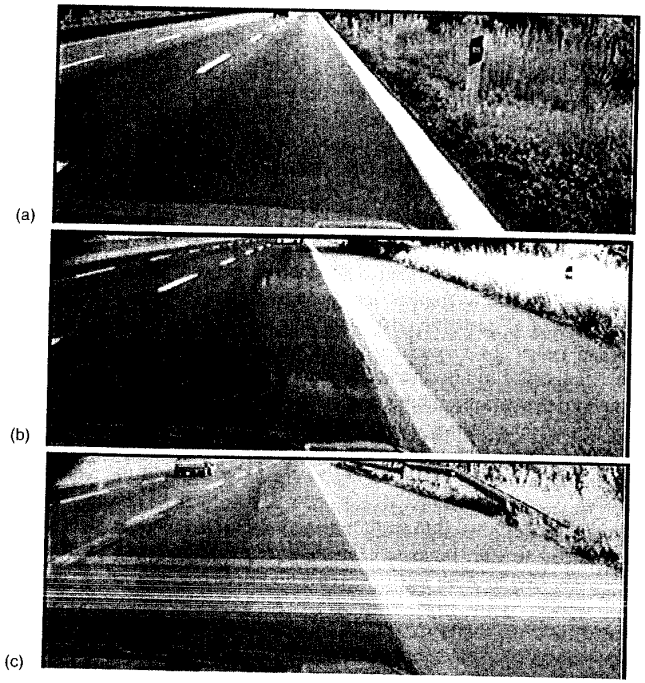


Fig. 14.45 The reflection problem: (*a*) image with no reflections; (*b*) image with a strong reflection; (c) oversaturated image due to reflections on the windscreen.

14.4.3 Statistical analysis of the tour

The analysis of the data collected during the tour allowed the computation of a number of statistics regarding system performance (see Table 14.2). In particular, for each stage of the tour the average and the maximum speed of the vehicle during automatic driving were computed. The average speed was strongly influenced by the heavy traffic conditions (especially on Turin's and Milan's by-passes) and by the presence of toll stations, junctions, and road works.

The automatic driving percentage and the maximum distance automatically driven show high values despite the presence of many tunnels (particularly during the Appennines routes Ancona–Rome and Florence–Bologna) and of several stretches of road with absent or worn lane markings (Ferrara–Ancona and Ancona–Rome) or even no lane markings at all (Florence–Bologna). It is fundamentally important also to note that some stages included passing through toll stations and transiting in by-passes with heavy traffic and frequent queues during which the system had to be switched off.

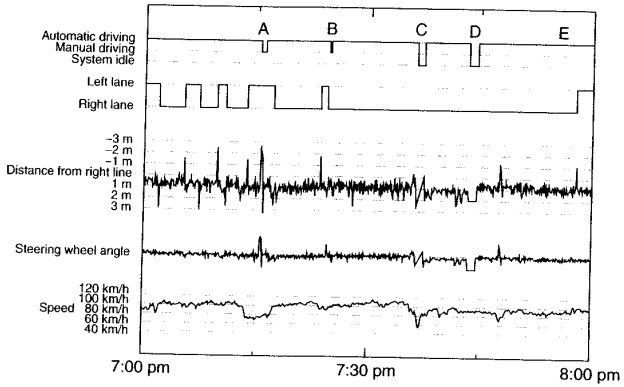
Table 14.2 Statistical data regarding the system performance during the tour

Stage	Date	Departure	Arrival	Kilometres	Average speed (km/h)	Maximum speed (km/h)	Percentage of automatic driving	Maximum distance in automatic (km)
1	1 June 1998	Parma	Turin	245	86.6	109	93.3	23.4
2	2 June 1998	Turin	Pavia	175	80.2	95	85.1	42.2
3	2 June 1998	Pavia	Ferrara	340	89.8	115	86.4	54.3
4	3 June 1998	Ferrara	Ancona	260	89.8	111	91.1	15.1
5	4 June 1998	Ancona	Rome	365	88.4	108	91.1	20.8
6	5 June 1998	Rome	Florence	280	87.5	110	95.4	30.6
7	6 June 1998	Florence	Parma	195	89.0	123	95.1	25.9

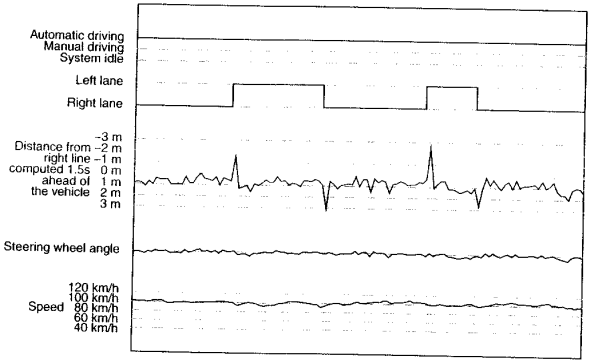
14.4.4 Detailed analysis of one hour of automatic driving

Figure 14.46 shows the system behaviour during a one hour period of time (from 7 pm to 8 pm on 2 June 1998) of the tour's third stage (approximately 90 km from Verona to Ferrara). Figure 14.46 shows the following from top to bottom:

- The selected functionality: during this period of time the system always worked in automatic mode except in four situations indicated with A, B, C and D, in which:
  (A) ARGO passed through two tunnels (near Vicenza);
  (B) the system at the end of an overtaking manoeuvre did not recognize the lane markings due to an occlusion;
  (C) and (D) the vehicle passed two working areas with no lane markings.
  In the first two cases, the system switched back to manual driving automatically and warned the driver; whilst in the other two cases the system was switched off manually. It is important to note that in correspondence to these four situations the logged data (apart from speed) is meaningless. The images dumped on disk on these occasions are shown in Figure 14.47.
- The vehicle position on the lane: ARGO made a number of lane changes, moving from the rightmost lane to the central lane and vice versa (ARGO was running on a three-lane highway).
- The distance between the right wheel and the right lane markings (in centimetres). It can be noted that during lane changes the distance alters abruptly; when the vehicle position is again central with respect to the new lane, the systems locks onto the new lane and the distance returns to the nominal value. The selected value of the distance is 80 cm.
- The steering wheel angle automatically chosen by the system. It is possible to note that this angle is proportional to the distance shown by the previous graph, with the only exception of the peaks due to lane changes.
- Finally, the speed of the vehicle. It is possible to note the low speed in the situations identified by C and E respectively, due to road works and driving on a junction (between A4 and A13, in which is also visible an evident movement of the steering wheel).



(a) Third stage: Pavia – Ferrara (2 June 1998)



(b) 10 minutes during the 3rd leg

Fig. 14.46 System behaviour during (*a*) one hour and (*b*) 10 minutes of automatic driving.

14.4.5 Discussion and current enhancements

This section summarizes the main problems that lowered the system performance. As mentioned above, the presence of tunnels and long bridges, when combined with strong sunshine, produces very contrasted shadows and sun spots. In order to adjust the value of the automatic gain control, the cameras compute the average brightness of the whole frame. This produces poor results when the scene is highly contrasted as shown in Figure 14.48.

Highly contrasted scenes were mainly found in the situations explained above (tunnels and bridges), but also when passing on different patches of asphalt (usually a new dark grey and an old light grey one), as shown in Figure 14.49. In the same

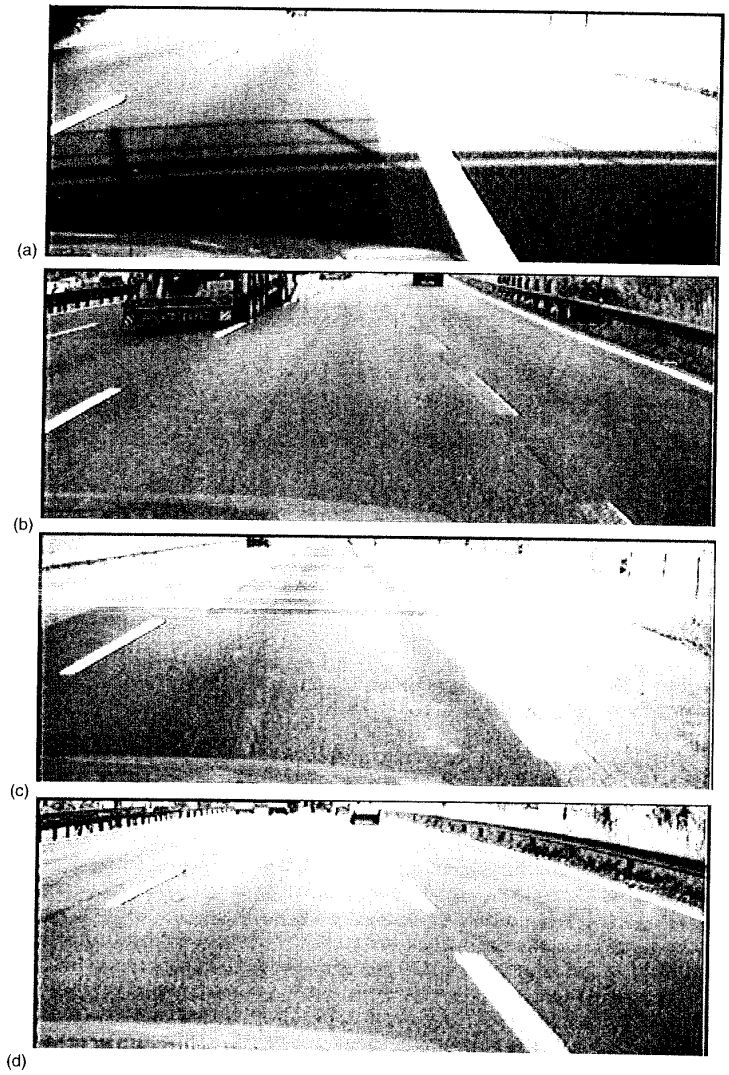


Fig. 14.47 Images dumped in the anomalous situations A, B, C, and D respectively.

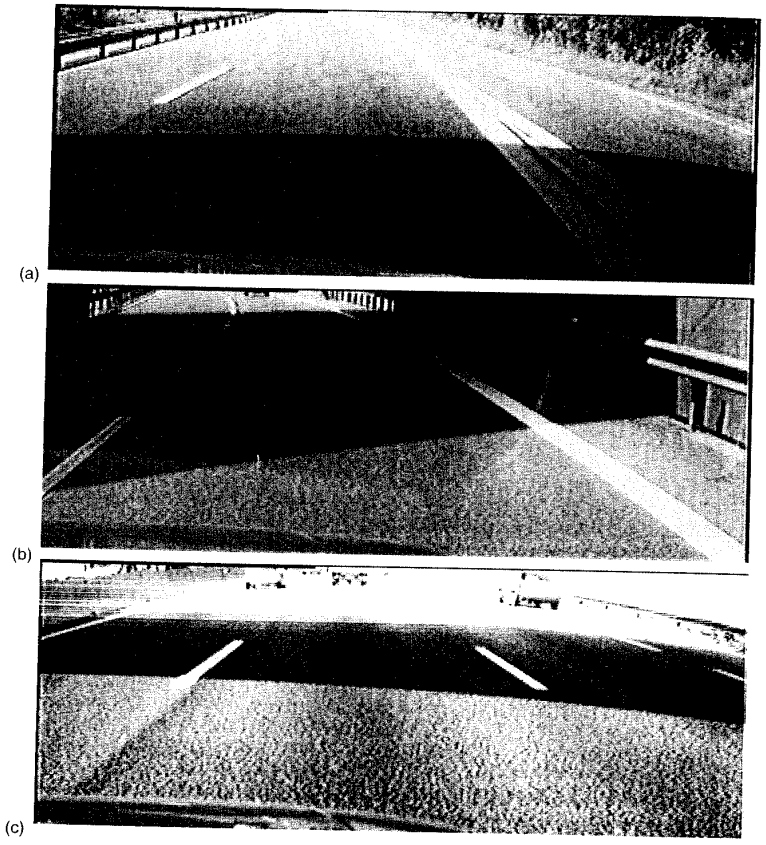


Fig. 14.48 The problem of high contrast: (*a*) exit from a tunnel; (*b*) long bridge; (*c*) short bridge.

figure a strong reflection within the internal windscreen surface – which produces a high image saturation – is also visible.

Figure 14.50 shows a situation in which the system had a major fault: a vehicle occluded the visibility of the right lane marking in correspondence to a highway exit. Since the only visible lane marking was the continuous line representing the exit, the system followed that line, therefore requiring a user intervention. Since the GOLD system does not include the possibility of automatically choosing whether to stay on the highway or to exit, a user intervention was always requested at each exit. This explains why the percentage of automatic driving was high even when the maximum distance driven in automatic was not that long (see Table 14.2). Other user interventions were required in correspondence to toll stations and road works.

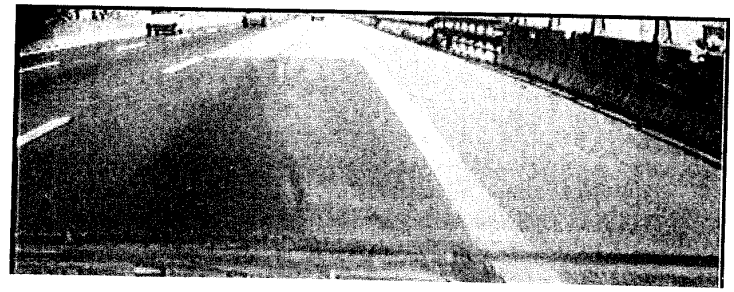


Fig. 14.49 The saturation problem: two different asphalt patches produce a highly contrasted scene which, along with a strong reflection on the windscreen, saturates the image.

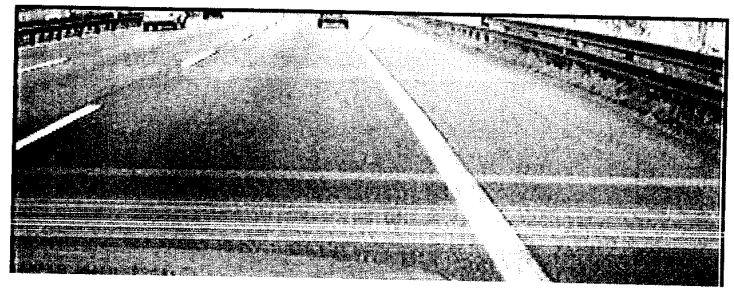


Fig. 14.50 Situation which caused a system fault: a vehicle occluded the visibility of the right lane marking in correspondence to an exit.

Another serious problem happened in a stretch of road just after the conclusion of road works: on the ground were still present the lines that were painted to delimit the diversion and, as a result, the vehicle followed those lines.

The system demonstrated high robustness with respect to the following: horizontal road signs (Figure 14.51(a)) and fog markings (Figure 14.51(b)); in presence of forks, junctions, and exits, as shown in Figure 14.51(c); heavy traffic conditions that were encountered, mainly on the Turin, Milan, and Rome by-passes, as shown in Figure 14.51(d); and the presence of the guard-rail in Figure 14.51(e).

Moreover, also high temperatures, different light conditions (the third stage of the tour terminated during the night), and high speeds (up to 120 km/h) did not influence the stability and robustness of the whole system, both for hardware and software. The vehicle only experienced a slightly oscillating behaviour in correspondence to sloping stretches of road.

Finally, a few minor problems experienced during the tour were: a power failure due to an overloading of the power supply, interference of an on-board cellular phone with the cameras and acquisition board, and some human errors in the use of the control panel.

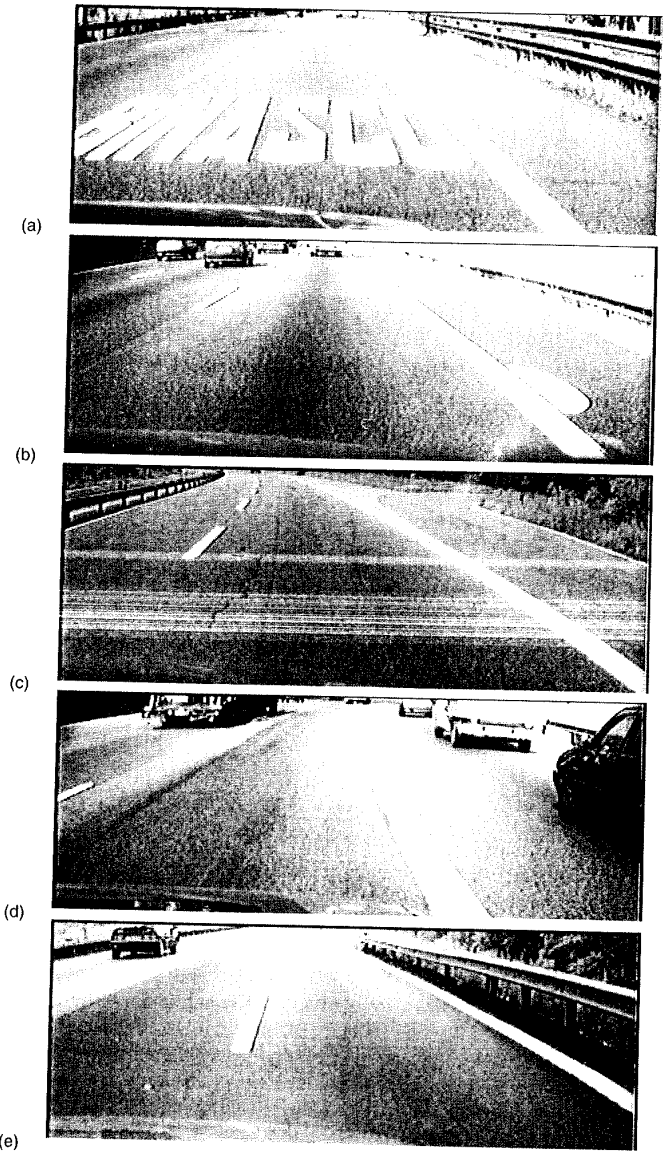


Fig. 14.51 Situations in which the system demonstrated good reliability: (*a*) horizontal road signs; (*b*) fog markings; (*c*) exits; (*d*) heavy traffic; (*e*) guard-rail.

These problems have been deeply analysed and the following solutions have been considered and implemented (as discussed in the previous sections).

- Lane detection based on two lines: when the visibility of one lane marker is low or missing, the system relies on the other one, reconstructing the missing information on the basis of historical information; the system can learn from past experience.
- New cameras (with higher dynamics and faster automatic gain control) have been considered and will be tested in the future.
- The new vehicle detection procedure has been developed, together with vehicle tracking.
- A new functionality – platooning – has been demonstrated, which allows ARGO to automatically follow any preceding vehicle.

References

Bertozzi, M. and Broggi, A. (1998). GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. *IEEE Trans. on Image Processing*, **7**(1), 62–81, January.

Bertozzi, M., Broggi, A. and Fascioli, A. (1998a). An extension to the Inverse Perspective Mapping to handle non-flat roads. *Proceedings of the IEEE Intelligent Vehicles Symposium '98*, pp. 305–10, Stuttgart, Germany, October.

Bertozzi, M., Broggi, A. and Fascioli, A. (1998b). Stereo Inverse Perspective Mapping: Theory and Applications. *Image and Vision Computing Journal*, **8**(16), pp. 585–90.

Broggi, A., Bertozzi, M., Fascioli, A. and Conte, G. (1999). *Automatic Vehicle Guidance: the Experience of the ARGO Vehicle*. World Scientific, April. ISBN 981–02–3720–0.

Fascioli, A. (2000). Vision-based Automatic Vehicle Guidance: Development and Test of a Prototype. PhD thesis, Dipartimento di Ingegneria dell'Informazione, Università di Parma, Italy, January.