GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection

Massimo Bertozzi, *Student Member, IEEE,* and Alberto Broggi, *Associate Member, IEEE*

***Abstract*—This paper describes the *Generic Obstacle and Lane Detection* system (GOLD), a stereo vision-based hardware and software architecture to be used on moving vehicles to increment road safety. Based on a full-custom massively parallel hardware, it allows to detect both generic obstacles (without constraints on symmetry or shape) and the lane position in a structured environment (with painted lane markings) at a rate of 10 Hz. Thanks to a geometrical transform supported by a specific hardware module, the perspective effect is removed from both left and right stereo images; the left is used to detect lane markings with a series of morphological filters, while both remapped stereo images are used for the detection of free-space in front of the vehicle. The output of the processing is displayed on both an onboard monitor and a control-panel to give visual feedbacks to the driver. The system was tested on the mobile laboratory (MOBLAB) experimental land vehicle, which was driven for more than 3000 km along extra-urban roads and freeways at speeds up to 80 km/h, and demonstrated its robustness with respect to shadows and changing illumination conditions, different road textures, and vehicle movement.**

I. Introduction

THE MAIN issues addressed in this work are *lane detection* and *obstacle detection*, both implemented using only visual data acquired from standard cameras installed on a mobile vehicle.

A. Lane Detection

Road following, namely the closing of the control loop that enables a vehicle to drive within a given portion of the road, has been differently approached and implemented in research prototype vehicles. Most of the systems developed worldwide are based on lane detection: first, the relative position of the vehicle with respect to the lane is computed, and then actuators are driven to keep the vehicle in a safe position. Others [15], [28], [38] are not based on the preliminary detection of the road position, but, as in the case of ALVINN [43], [44], derive the commands to issue to the actuators (steering wheel angles) directly from visual patterns detected in the incoming images. In any case, the knowledge of the lane position can be of use for other purposes, such as the determination of the regions of interest for other driving assistance functions.

The main problems that must be faced in the detection of road boundaries or lane markings are: 1) the presence of shadows, producing artifacts onto the road surface, and thus altering its texture, and 2) the presence of other vehicles on the path, partly occluding the visibility of the road. Although some systems have been designed to work on nonstructured roads (without painted lane markings) [28] or on unstructured terrain [39], [52], generally lane detection relies on the presence of painted road markings on the road surface. Therefore, since lane detection is generally based on the localization of a *specific pattern* (the lane markings) in the acquired image, it can be performed with the analysis of a *single* still image. In addition, some assumptions can aid the detection algorithm and/or speed-up the processing. They range from the analysis of specific regions of interest in the image (in which, due to both physical and continuity constraints, it is more probable to find the lane markings) [18] to the assumption of a fixed-width lane (thus dealing with only parallel lane markings), to the assumption of a precise road geometry (such as a clothoid) [18], [33], [58], to the assumption of a flat road (the one considered in this work).

The techniques implemented in the previously mentioned systems range from the determination of the characteristics of painted lane markings [30] eventually aided by color information [19] to the use of deformable templates (such as LOIS [31], DBS [7], or ARCADE [29]), to an edge-based recognition using a morphological paradigm [3], [5], [59], to a model-based approach (as implemented in VaMoRs [26] or SCARF [17]). A model-based analysis of road markings has also been used to perform the analysis of intersections in city traffic images [21], [32]; nevertheless, as discussed in [46], the use of a model-based search approach has several drawbacks, such as the problem of using and maintaining an appropriate geometrical road model, the difficulty in detecting and matching complex road features, and the complexity of the computations involved.

Moreover, some systems (such as [46]) work in the *velocity* domain instead of the *image* domain, thus using optical-flow techniques in order to minimize the horizontal relative movement of the lane markings with respect to the vehicle. Unfortunately, such a solution requires both the preliminary detection of lane markings and the following computation of the optical flow field.

B. Obstacle Detection

The techniques used in the detection of obstacles may vary according to the definition of "obstacle." If "obstacle" means

Manuscript received April 5, 1996; revised March 24, 1997. This work was supported in part by the Italian National Research Council under the framework of the Progetto Finalizzato Trasporti 2. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jeffrey J. Rodriguez.

The authors are with the Department of Information Technology, University of Parma, I-43100 Parma, Italy (e-mail: bertozzi@CE.UniPR.IT; broggi@CE.UniPR.IT).

Publisher Item Identifier S 1057-7149(98)00313-3.

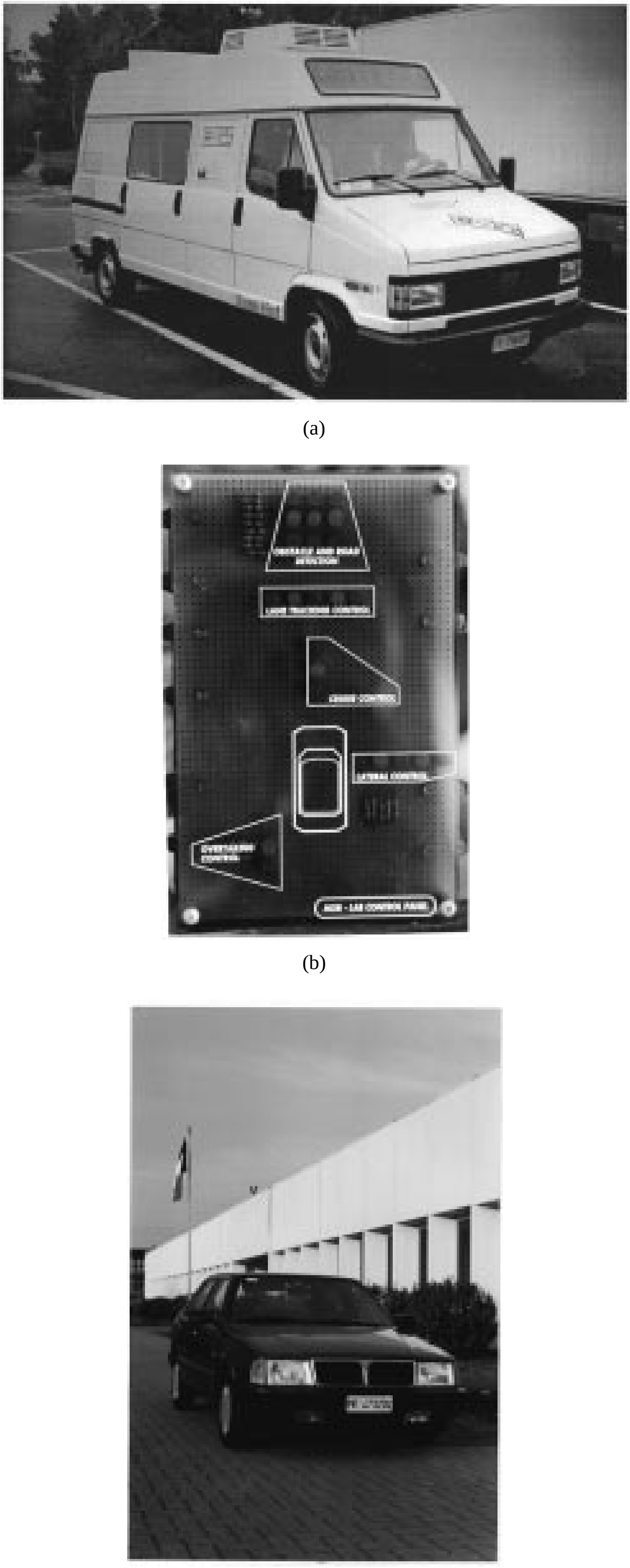
a vehicle, then the detection is based on a search for specific patterns, possibly supported by other features, such as shape [56], symmetry [61], or the use of a bounding box [1]. Also, in this case, the processing can be based on the analysis of a single still image.

Conversely, if we intend as obstacle any object that can obstruct the vehicle's driving path or anything raising out significantly from the road surface, obstacle detection is generally reduced to the detection of *free-space* instead of the recognition of specific patterns. In this case, different techniques can be used, such as 1) the analysis of the optical flow field, and 2) the processing of stereo images; both of these require two or more images, thus leading to a higher computational complexity, which is further increased by the necessity to handle noise caused by vehicle movements. Obstacle detection using the *optical flow approach* [13], [20] is generally divided into two steps: first, ego-motion is computed from the analysis of optical flow [25] or obtained from odometry [35]; then obstacles are detected by the analysis of the differences between the expected and the real velocity field.

On the other hand, the main problem of *stereo vision techniques* is the detection of correspondences between two stereo images (or three images, in case of trinocular vision [49]). The advantage of the analysis of stereo images instead of a monocular sequence of images is the possibility to detect directly the presence of obstacles, which, in case of an optical flow-based approach, is indirectly derived from the analysis of the velocity field. Moreover, in a limit condition where both vehicle and obstacles have small or null speeds, the second approach fails while the former still can detect obstacles. Furthermore, to decrease the intrinsic complexity of stereo vision, some domain specific constraints are generally adopted.

As in [33], the *Generic Obstacle and Lane Detection* (GOLD) system addresses both lane detection and obstacle detection at the same time: lane detection is based on a pattern-matching technique that relies on the presence of road markings, while the localization of obstacles in front of the vehicle is performed by the processing of pairs of stereo images: in order to be fast and robust with respect to camera calibration and vehicle movements, the detection of a *generic obstacle* is reduced to the determination of the *free-space* in front of the vehicle without any three-dimensional (3-D) world reconstruction.

Both functionalities share the same underlying approach (*image warping*), which is based on the assumption of a flat road. Such a technique has been successfully used for the computation of the optical flow field [36], for the detection of obstacles in a structured environment [34], [60], or in the automotive field [37], [42], [45] (using standard cameras) or [50], [57] (using linear cameras). It is based on a transform that, given a model of the road in front of the vehicle (e.g. flat road), remaps the right image onto the left; any disparity is caused by a deviation from the road model, thus detecting possible obstacles.



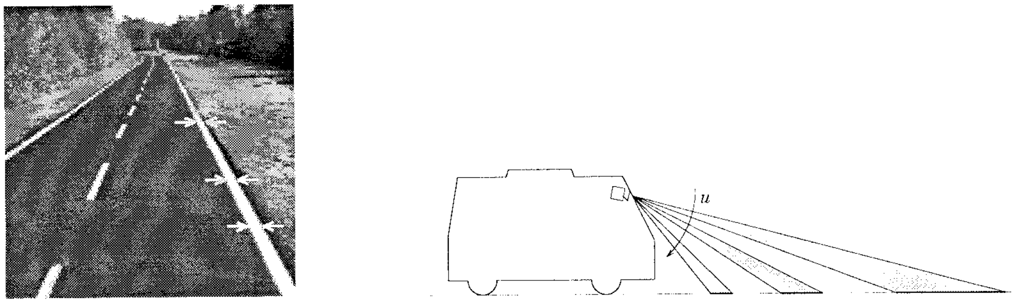
(a)

(b)

(c)

Fig. 1. (a) MOB-LAB land vehicle. (b) Control panel used as output to display the processing results. (c) ARGO autonomous passengers car.

Contrary to other works [33], [37], [42], GOLD performs two warpings instead of one, remapping both images into a different domain (road domain), in which the following processings are extremely simplified. Hence, the reprojection



(a) (b)

Fig. 2. (a) Road markings width changes according to their position within the image. (b) Due to the perspective effect, different pixels represent different portions of the road.

[33], [58] of the results in the road domain is no more required. Moreover, since both GOLD functionalities are based on the processing of images remapped into the same domain, the fusion of the result of the two independent processings is straightforward.

The GOLD system has been tested on mobile laboratory (MOB-LAB) experimental land vehicle, integrating the results of the Italian Research Units involved in the PROMETHEUS project. MOB-LAB [see Fig. 1(a)] is equipped with four cameras, two of which are used for this experiment, several computers, monitors, and a control-panel [see Fig. 1(b)] to give a visual feedback and warnings to the driver. The GOLD system is now being ported to ARGO [2] [see Fig. 1(c)], a Lancia Thema passenger car with automatic steering capabilities.

This work is organized as follows: Section II presents the basics of the underlying approach used to remove the perspective effect from a monocular image, while Section III describes its application to the processing of stereo images. Section IV describes the lane detection and obstacle detection functionalities; Section V presents the computing engine that has been developed as a support to the GOLD system; Section VI presents the analysis of the time performance of the current implementation; finally, Section VII ends the paper with a discussion about the problems of the system, their possible solutions, and future developments.

II. Inverse Perspective Mapping

Due to its intrinsic nature, low-level image processing is efficiently performed on single instruction multiple data (SIMD) systems by means of a massively parallel computational paradigm. Anyway, this approach is meaningful in the case of generic filterings (such as noise reduction, edge detection, and image enhancement), which consider the image as a mere collection of pixels, independent of their semantic content.

On the other hand, the implementation of more sophisticated filters requires some semantic knowledge. As an example, let us consider the specific problem of road markings detection in an image acquired from a vehicle. Due to the perspective effect introduced by the acquisition conditions, the road markings width changes according to their distance from the camera [see Fig. 2(a)]. Therefore, the correct detection of road markings should be based on matchings with patterns with different size, according to the specific position within the image. Unfortunately, this differentiated low-level processing cannot be efficiently performed on SIMD massively parallel systems, which by definition perform the *same* processing on each pixel of the image.

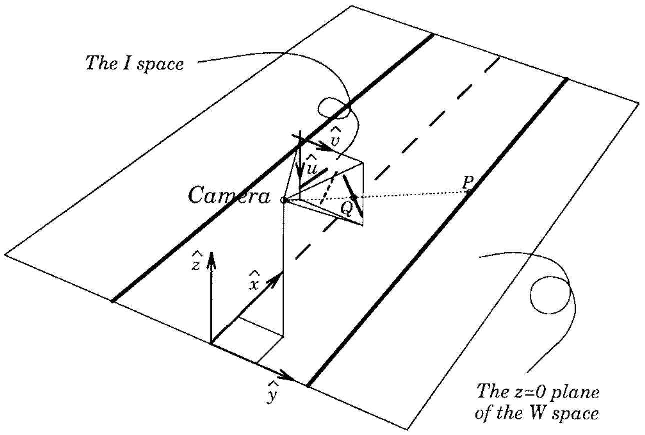
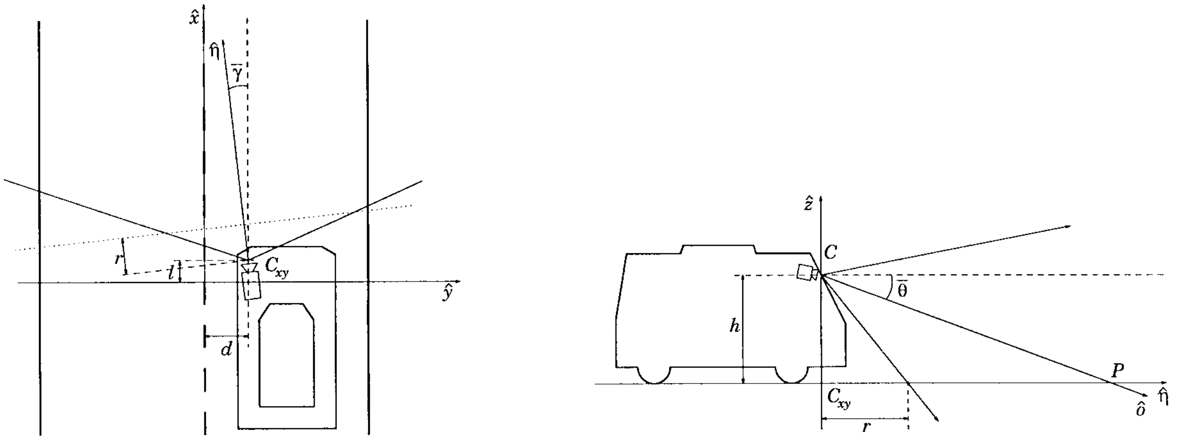


Fig. 3. Relationship between the two coordinate systems.

The perspective effect associates different meanings to different image pixels, depending on their position in the image [see Fig. 2(b)]. Conversely, after the removal of the perspective effect, each pixel represents the same portion of the road,[1] allowing a homogeneous distribution of the information among all image pixels; to remove the perspective effect, it is necessary to know the specific acquisition conditions (camera position, orientation, optics, etc.) and the scene represented in the image (the road, which is now assumed to be *flat*). This constitutes the *a priori* knowledge.

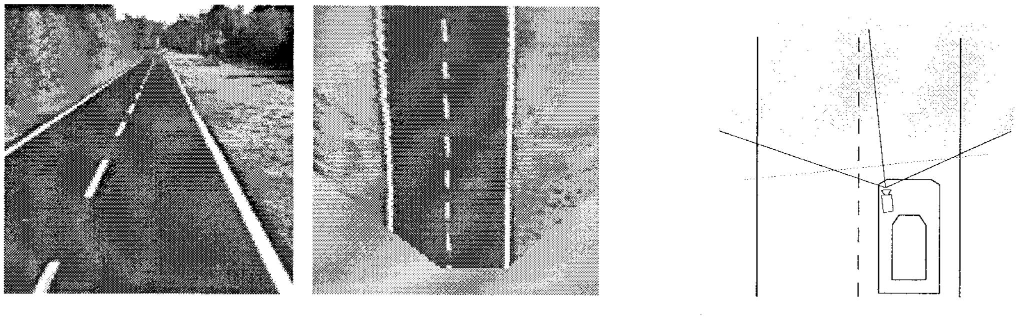
Now, recalling the example of road markings detection, the size and shape of the matching template can be independent of the pixel position. Therefore, road markings detection can be conveniently divided into two steps: the first, exploiting the *a priori* knowledge, is a transform that generates an image in a new domain where the detection of the features of interest is extremely simplified; the second, exploiting the sensorial

[1] A pixel in the lower part of the image of Fig. 2(a) represents a few cm^2 of the road, while a pixel in the middle of the same image represents a few tens of cm^2, or even more.



(a) (b)

Fig. 4. (a) The xy plane in the $\mathcal{W}$ space and (b) the $z\eta$ plane.



(a) (b)

Fig. 5. (a) Original and remapped images. (b) In grey, the visible portion of the road.

data, consists of a mere low-level morphological processing. The removal of the perspective effect allows to detect road markings through an extremely simple and fast morphological processing that can be efficiently implemented on massively parallel SIMD architectures.

A. Removing the Perspective Effect

The procedure aimed to remove the perspective effect resamples the incoming image, remapping each pixel toward a different position and producing a new two-dimensional (2-D) array of pixels. The resulting image represents a top view of the road region in front of the vehicle, as it was observed from a significant height.

Two Euclidean spaces are defined, as follows.

- $\mathcal{W} = \{(x, y, z)\} \in E^3$, representing the 3-D world space (*world-coordinate*), where the real world is defined.
- $\mathcal{I} = \{(u, v)\} \in E^2$, representing the 2-D image space (*screen-coordinate*), where the 3-D scene is projected.

The image acquired by the camera belongs to the $\mathcal{I}$ space, while the remapped image is defined as the $z = 0$ plane of the $\mathcal{W}$ space (according to the assumption of a flat road). The remapping process projects the acquired image onto the $z = 0$ plane of the 3-D world space $\mathcal{W}$. Fig. 3 shows the relationships between the two spaces $\mathcal{W}$ and $\mathcal{I}$.

1) $\mathcal{I} \rightarrow \mathcal{W}$ Mapping: In order to generate a 2-D view of a 3-D scene, the following parameters must be known [41].

1) *Viewpoint:* camera position is $C = (l, d, h) \in \mathcal{W}$.
2) *Viewing Direction:* optical axis $\hat{o}$ is determined by the following angles:
   - $\overline{\gamma}$: the angle formed by the projection (defined by versor $\hat{\eta}$) of the optical axis $\hat{o}$ on the plane $z = 0$ and the x axis [as shown in Fig. 4(a)];
   - $\overline{\theta}$: the angle formed by the optical axis $\hat{o}$ and versor $\hat{\eta}$ [as shown in Fig. 4(b)].
3) *Aperture:* camera angular aperture is 2α.
4) *Resolution:* camera resolution is $n \times n$.

After simple manipulations [6], the final mapping $f\colon \mathcal{I} \rightarrow \mathcal{W}$ as a function of u and v is given by

$$\begin{aligned} x(u,v) &= h \times \operatorname{ctg}\left[(\overline{\theta} - \alpha) + u\frac{2\alpha}{n-1}\right] \\ &\quad \times \cos\left[(\overline{\gamma} - \alpha) + v\frac{2\alpha}{n-1}\right] + l \\ y(u,v) &= h \times \operatorname{ctg}\left[(\overline{\theta} - \alpha) + u\frac{2\alpha}{n-1}\right] \\ &\quad \times \sin\left[(\overline{\gamma} - \alpha) + v\frac{2\alpha}{n-1}\right] + d \\ z &= 0 \end{aligned} \tag{1}$$

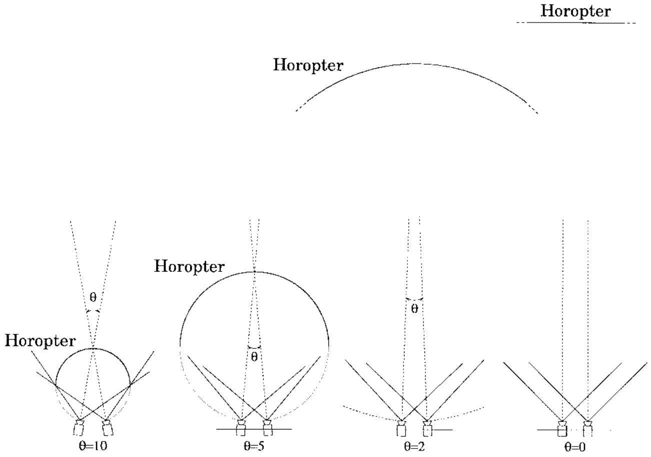


Fig. 6. Horopter surface corresponding to different angles between the optical axes of two stereo cameras.

with $u, v = 0, 1, \cdots, n-1$. Given the coordinates (u, v) of a generic point Q in the $\mathcal{I}$ space, (1) return the coordinates $(x, y, 0)$ of the corresponding point P in the $\mathcal{W}$ space (see Fig. 3).

2) $\mathcal{W} \rightarrow \mathcal{I}$ Mapping: The inverse transform g: $\mathcal{W} \rightarrow \mathcal{I}$ (the dual mapping) is given as follows [6]:

$$u(x,y,0) = \frac{\mathrm{arctg}\left[\dfrac{h \sin \gamma(x,y,0)}{y-d}\right] - (\overline{\theta} - \alpha)}{\dfrac{2\alpha}{n-1}} \quad \text{and}$$

$$v(x,y,0) = \frac{\mathrm{arctg}\left[\dfrac{y-d}{x-l}\right] - (\overline{\gamma} - \alpha)}{\dfrac{2\alpha}{n-1}}. \tag{2}$$

The remapping process defined by (2) removes the perspective effect and recovers the texture of the $z = 0$ plane of the $\mathcal{W}$ space. It is implemented scanning the array of pixels of coordinates $(x, y, 0) \in \mathcal{W}$ which form the remapped image, in order to associate to each of them the corresponding value assumed by the point of coordinates $(u(x,y,0), v(x,y,0)) \in \mathcal{I}$.

As an example, Fig. 5(a) shows the original and remapped images: it is clearly visible that in this case the road markings width is almost invariant within the whole image. The resolution of the remapped image has been chosen as a trade-off between information loss and processing time; the remapped image shown in Fig. 5(a) has been obtained without preserving the original aspect-ratio. Note that the lower portion of the remapped image is undefined: this is due to the specific camera position and orientation [see Fig. 5(b)].

III. Stereo Inverse Perspective Mapping

A 3-D description of the world using a single 2-D image is impossible without *a priori* knowledge, due to the depth loss during acquisition; for many years stereo vision has been investigated as an answer to this problem. Generally, *traditional* techniques for the processing of pairs of stereo images are divided into the following four main steps:

1) calibration of the two cameras;
2) localization of a feature in an image;
3) identification and localization of the same feature in the other image;
4) reconstruction of the 3-D scene.

Whenever the mapping between points corresponding to the same feature (*homologous points*) can be determined, the problem of 3-D reconstruction can be solved using triangulations. The intrinsic complexity of the determination of homologous points can be reduced with the introduction of some domain-specific constraints, such as the assumption of a flat road in front of the cameras.

The set of points $\mathcal{H} = \{(x, y, z) | u^{(L)} = u^{(R)} \wedge v^{(L)} = v^{(R)}\}$, where $(u^{(L)}, v^{(L)})$ and $(u^{(R)}, v^{(R)})$ represent the projection of (x, y, z) in the $\mathcal{I}$ space of the left and right camera respectively, is called *horopter* and represents the zero disparity surface of the stereo system [11]. This means that the two stereo views of an object whose shape and displacement matches the horopter are identical. This concept is extremely useful when the horopter coincides with a model of the road surface, since any deviation from this model can be easily detected. The horopter is a spherical surface, the smaller the difference between the orientation of the two cameras (*camera vergence*) the larger the radius [22]. Assuming a small camera vergence, as generally happens in the automotive field, the horopter can be considered planar. As shown in Fig. 6, the horopter can be moved acting on camera vergence parameters. Unfortunately, the horopter cannot be overlapped with the $z = 0$ plane (representing the flat road model) using only camera vergence; for this purpose, *electronic vergence*, such as inverse perspective mapping (IPM), is required.

In this way the search for homologous points is reduced to a simple verification (check) of the shape of the horopter: in fact under the flat road hypothesis, the IPM algorithm can be used to produce an image representing the road as seen from the top. Using the IPM algorithm with appropriate parameters on stereo images, different patches of the road surface can be obtained. Moreover the knowledge of the parameters of the whole vision system allows to bring the two road patches to correspondence. This means that, under the flat road hypothesis, pairs of pixels having the same image coordinates in the two remapped images are homologous points and represent the same points in the road plane.

The flat road hypothesis can be verified computing the difference between the two remapped images: a *generic obstacle* (anything raising out from the road) is detected if the difference image presents sufficiently large clusters of nonzero pixels having a specific shape. Due to the different position of the two cameras, the difference image can be computed only for the overlapping area of the two road patches.

In addition, it is easily demonstrable that the IPM algorithm maps straight lines perpendicular to the road plane into straight lines passing through the projection $C_{xy} \equiv (l, d, 0)$ of the camera onto the plane $z = 0$ (see Fig. 4): using formula (1), a vertical straight line is represented by the set of pixels

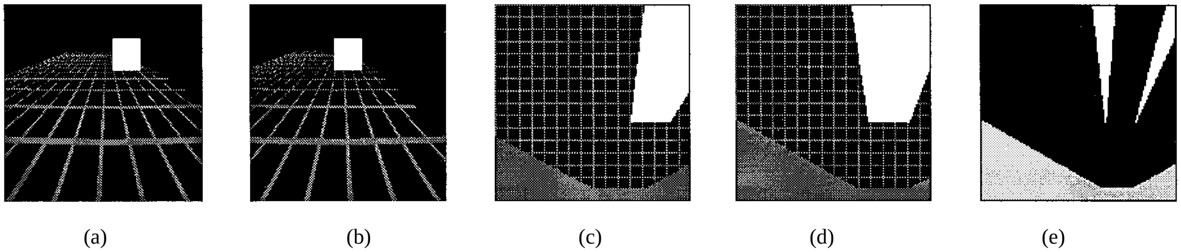


Fig. 7. Homogeneous ideal object. (a) Left view. (b) Right view. (c) Remapped left view. (d) Remapped right view. (e) Difference between (c) and (d) showing in light grey the area not seen by both cameras.

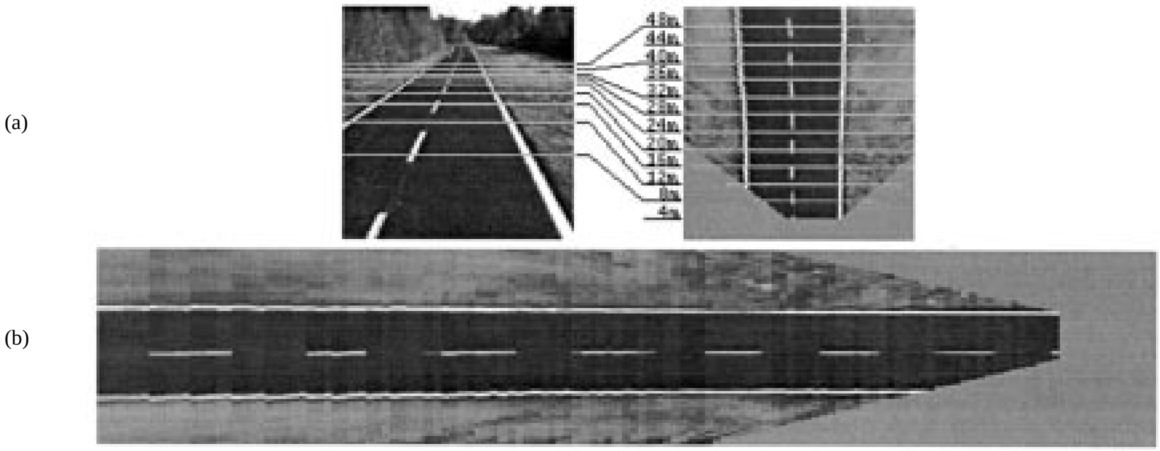


Fig. 8. (a) Horizontal calibration of the MOB-LAB vision system. (b) Rotated version of the remapped image considering an aspect ratio of 1 : 1.

$\{P(u_i, v_i) | v_i = \overline{v} = constant\}$; thus, defining

$$k_{\overline{v}} \triangleq (\overline{\gamma} - \alpha) + \overline{v}\frac{2\alpha}{n-1} = constant \quad \text{and}$$
$$p(u) \triangleq \tan\left[(\overline{\theta} - \alpha) + u\frac{2\alpha}{n-1}\right] \tag{3}$$

it follows that

$$\begin{aligned} x(u, \overline{v}) &= \frac{h}{p(u)} \cos k_{\overline{v}} + l; \\ y(u, \overline{v}) &= \frac{h}{p(u)} \sin k_{\overline{v}} + d; \\ z &= 0 \end{aligned} \tag{4}$$

which represents the straight line $x = (y - d)\ \mathrm{ctg}(k_{\overline{v}}) + l$ passing through C_{xy}.

Thus, the vertical edges of a generic obstacle are mapped into two straight lines intersecting in C_{xy}. Due to the different coordinates of the projection of the two cameras, in a ideal case the difference between the two remapped images presents two triangles. This behavior is shown in Fig. 7, 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 showing also the overlapping between the two viewing areas.

A. Camera Calibration

From the above description, it can be seen that the calibration of the vision system plays a basic role. Recalling the definitions and notations given in Section II-A1, the calibration parameters can be divided into the following two categories.

- *Intrinsic parameters* (camera angular aperture 2α and camera resolutions $n \times n$), which are fixed.
- *Extrinsic parameters* (view point and viewing direction), which can be determined by measurements and possibly tuned.

Two of the four extrinsic parameters (h and $\overline{\theta}$) are in common between the two cameras, while the other two (d and $\overline{\gamma}$) are determined by the analysis of input images. After the independent calibration of both cameras, a fine tuning of the $d^{(L)}, d^{(R)}, \overline{\gamma}^{(L)}$ and $\overline{\gamma}^{(R)}$ parameters is obtained applying the stereo IPM algorithm iteratively, and minimizing the disparities between the two remapped images of a flat road acquired with the vehicle standing still.

TABLE I
ACQUISITION PARAMETERS OF THE CAMERA INSTALLED ONTO MOB-LAB

		Left Camera	**Right Camera**
Intrinsic Parameters	2α	28°00′	
	n	512	
Extrinsic Parameters	h	200 cm	
	$\overline{\theta}$	8°55′	
	d	95 cm	185 cm
	$\overline{\gamma}$	0°45′	0°00′

The acquisition parameters of the camera installed onto MOB-LAB are shown in Table I.

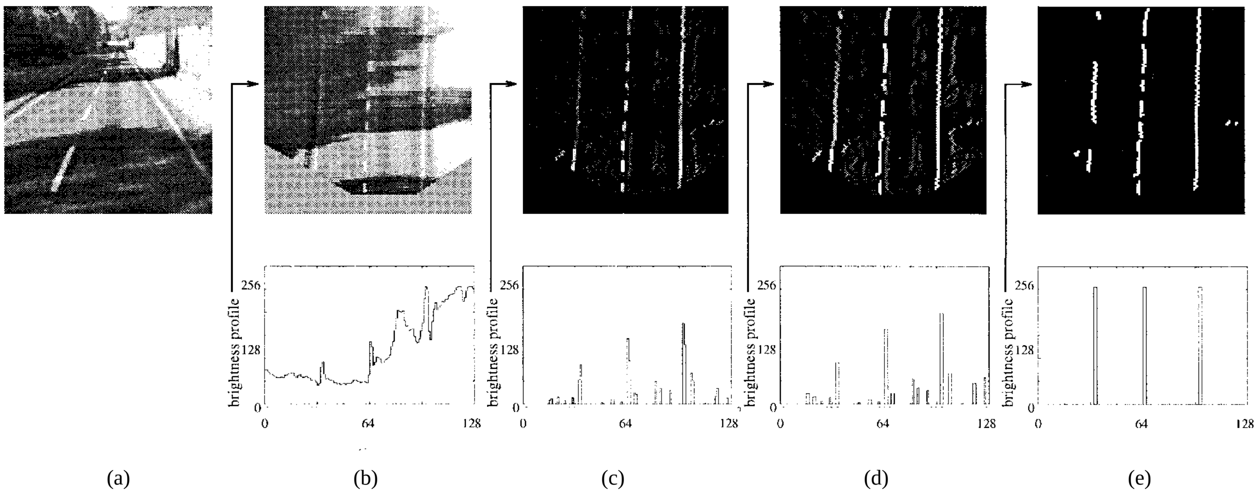
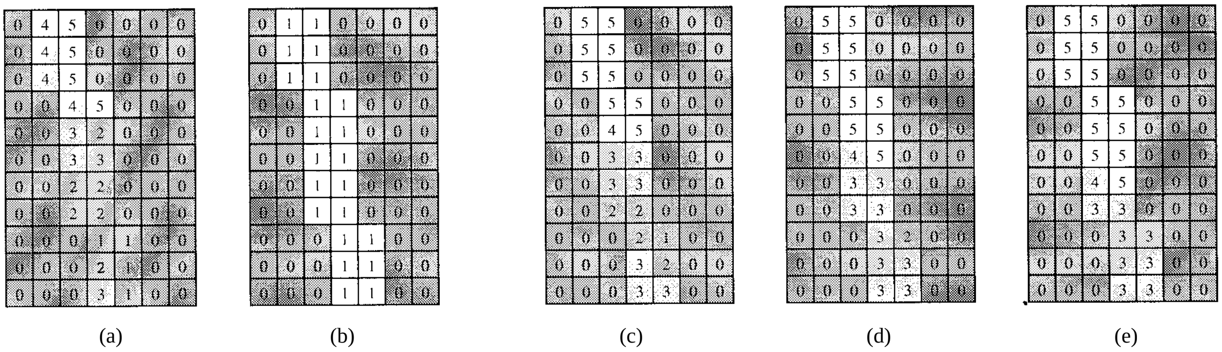


Fig. 9. Sequence of images produced by the low-level lane detection phase together with a linear profile of one of their lines. (a) Original. (b) Remapped. (c) Filtered. (d) Enhanced. (e) Binarized.



(a)

0	4	5	0	0	0	0
0	4	5	0	0	0	0
0	4	5	0	0	0	0
0	0	4	5	0	0	0
0	0	3	2	0	0	0
0	0	3	3	0	0	0
0	0	2	2	0	0	0
0	0	2	2	0	0	0
0	0	0	1	1	0	0
0	0	0	2	1	0	0
0	0	0	3	1	0	0

(b)

0	1	1	0	0	0	0
0	1	1	0	0	0	0
0	1	1	0	0	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	0	1	1	0	0
0	0	0	1	1	0	0
0	0	0	1	1	0	0

(c)

0	5	5	0	0	0	0
0	5	5	0	0	0	0
0	5	5	0	0	0	0
0	0	5	5	0	0	0
0	0	4	5	0	0	0
0	0	3	3	0	0	0
0	0	3	3	0	0	0
0	0	2	2	0	0	0
0	0	0	2	1	0	0
0	0	0	3	2	0	0
0	0	0	3	3	0	0

(d)

0	5	5	0	0	0	0
0	5	5	0	0	0	0
0	5	5	0	0	0	0
0	0	5	5	0	0	0
0	0	5	5	0	0	0
0	0	4	5	0	0	0
0	0	3	3	0	0	0
0	0	3	3	0	0	0
0	0	0	3	2	0	0
0	0	0	3	3	0	0
0	0	0	3	3	0	0

(e)

0	5	5	0	0	0	0
0	5	5	0	0	0	0
0	5	5	0	0	0	0
0	0	5	5	0	0	0
0	0	5	5	0	0	0
0	0	5	5	0	0	0
0	0	4	5	0	0	0
0	0	3	3	0	0	0
0	0	0	3	3	0	0
0	0	0	3	3	0	0
0	0	0	3	3	0	0

Fig. 10. Geodesic morphological dilation used for image enhancement. (a) Input image. (b) Control image. (c)–(e) Results of the first three iterations.

Fig. 8(a) shows the horizontal calibration of the left camera installed onto MOB-LAB, while Fig. 8(b) shows the remapped image with an aspect ratio of one.

Since during the tests the acquisition system was used also for other experiments, no other cameras setups were tested. Section VII presents a brief discussion about benefits in varying the camera setup and about the impact of drifts due to vehicle movements.

IV. Driving Assistance Functions

In the following section the lane detection and obstacle detection functionalities are discussed. Both of them are divided into a low-level phase that can be efficiently expressed with a SIMD computational paradigm and a serial high- and medium-level phase.

A. Lane Detection

Lane detection is performed assuming that a road marking in the $z = 0$ plane of the $\mathcal{W}$ space (i.e., in the remapped image) is represented by a quasivertical bright line of constant width surrounded by a darker region (the road). Thus, the pixels belonging to a road marking have a brightness value higher than their left and right neighbors at a given horizontal distance. Fig. 9(a) and (b) shows the original and remapped images together with a linear brightness profile of a horizontal line.

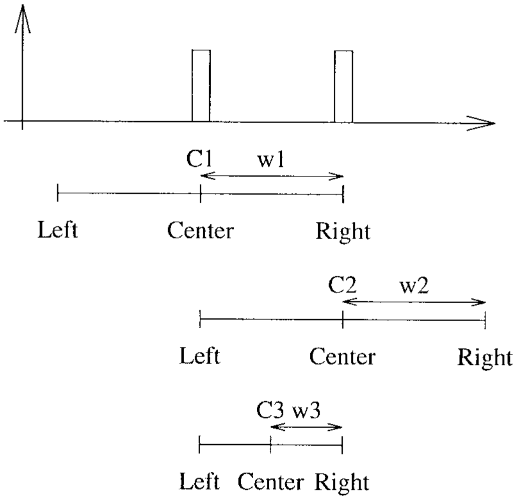


Fig. 11. Two nonzero pixels can represent three different road configurations.

The first phase of road markings detection is thus based on a linewise determination of horizontal black-white-black transitions, while the following medium-level processing is aimed

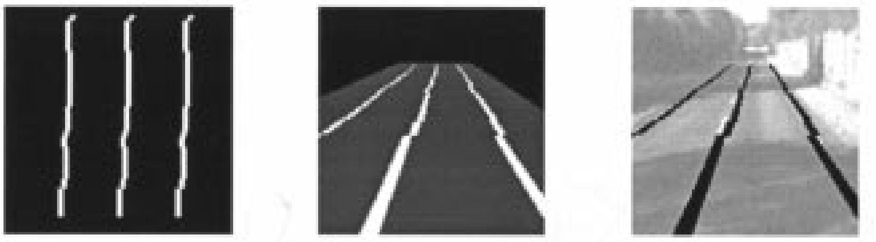


Fig. 12. Results of the medium level processing. (a) Reconstruction of the road markings. (b) Reintroduction of the perspective effect. (c) Superimposition of the previous result onto a brighter version of the original image.

to the extraction of the information and the reconstruction of road geometry.

1) Parallel Feature Extraction: The brightness value $b(x,y)$ of a generic pixel of the remapped image is compared to its horizontal left and right neighbors 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:

$$r(x,y)=\begin{cases} d_{+m}(x,y)+d_{-m}(x,y), \\ \quad \text{if } (d_{+m}(x,y)>0)\wedge(d_{-m}(x,y)>0) \\ 0, \quad \text{otherwise} \end{cases} \tag{5}$$

where

$$\begin{aligned} d_{+m}(x,y) &= b(x,y)-b(x,y+m) \\ d_{-m}(x,y) &= b(x,y)-b(x,y-m) \end{aligned} \tag{6}$$

represent the horizontal brightness gradient. The choice of m depends on the road markings width, on the image acquisition process, and on the parameters used in the remapping phase. Considering $m=2$, the filtered image is shown in Fig. 9(c).

Due to different illumination conditions (e.g., in presence of shadows), the road markings may have different brightness, yet maintain their superiority relationship with their horizontal neighbors. Thus, 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* [53] with the following binary structuring element:

	●	
●	●	●
	●	

where

$$c(x,y)=\begin{cases} 1, & \text{if } r(x,y)\neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

is the *control image* [53]. The result of the geodesic dilation is the product between the control image and the maximum value computed among all the pixels belonging to the neighborhood described by the structuring element. The iterative application of the geodesic dilation corresponds to the widening of the neighborhood, except in the directions in which the control values are zero. Fig. 10 shows the results of three iterations on a portion of an image that represents a lane marking.

Some results of lane detection in a number of different conditions are shown in Fig. 13.

Moreover, since

$$r(x,y)\neq 0 \Longrightarrow \begin{cases} r(x,y-m)=0 \\ r(x,y+m)=0, \end{cases} \tag{8}$$

the pixels of the filtered image at a distance m from a road marking assume a zero value; according to (7) their control value is zero and, thus, they form a barrier to the propagation of the maximum value. The enhanced image (after eight iterations) is shown in Fig. 9(d).

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

$$t(x,y)=\begin{cases} 1, & \text{if } e(x,y)\geq \dfrac{m(x,y)}{k} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $e(x,y)$ represents the enhanced image, $m(x,y)$ the maximum value computed in a given $c \times c$ neighborhood, and k is a constant. The result of the binarization of Fig. 9(d), considering $k=2$ and $c=7$, is presented in Fig. 9(e).

2) Feature Identification: The goal of the successive medium-level processing is the determination of the geometry of the road, starting from the thresholded image. This binary image is scanned row by row and for each line i all its nonzero pixels are considered in pairs: a nonzero pixel can represent the left edge, the right edge, or the center line; Fig. 11 shows the three different road configurations that a pair of nonzero pixels can represent, considering the road as identified by three nonzero pixels. Each one of these three possible configurations produces a pair (c_i, w_i), where c_i represents the coordinate of the road medial-axis and w_i represents its corresponding lane width. Anyway, not all pairs (c_i, w_i) correspond to a valid road configuration, and only the ones satisfying the following constraints will be considered: $0 \leq c_i \leq N; w_i < (N/3); c_i - w_i \leq \frac{3}{4}N; c_i + w_i \geq (N/4)$; where N is the image horizontal size.

A histogram is built considering the values assumed by w_i for each line of the image; the peak of the histogram corresponds to the more frequent value of lane width. In order to allow a nonfixed road geometry (and also the handling of curves) the histogram is lowpass filtered; finally, its maximum value W is determined.

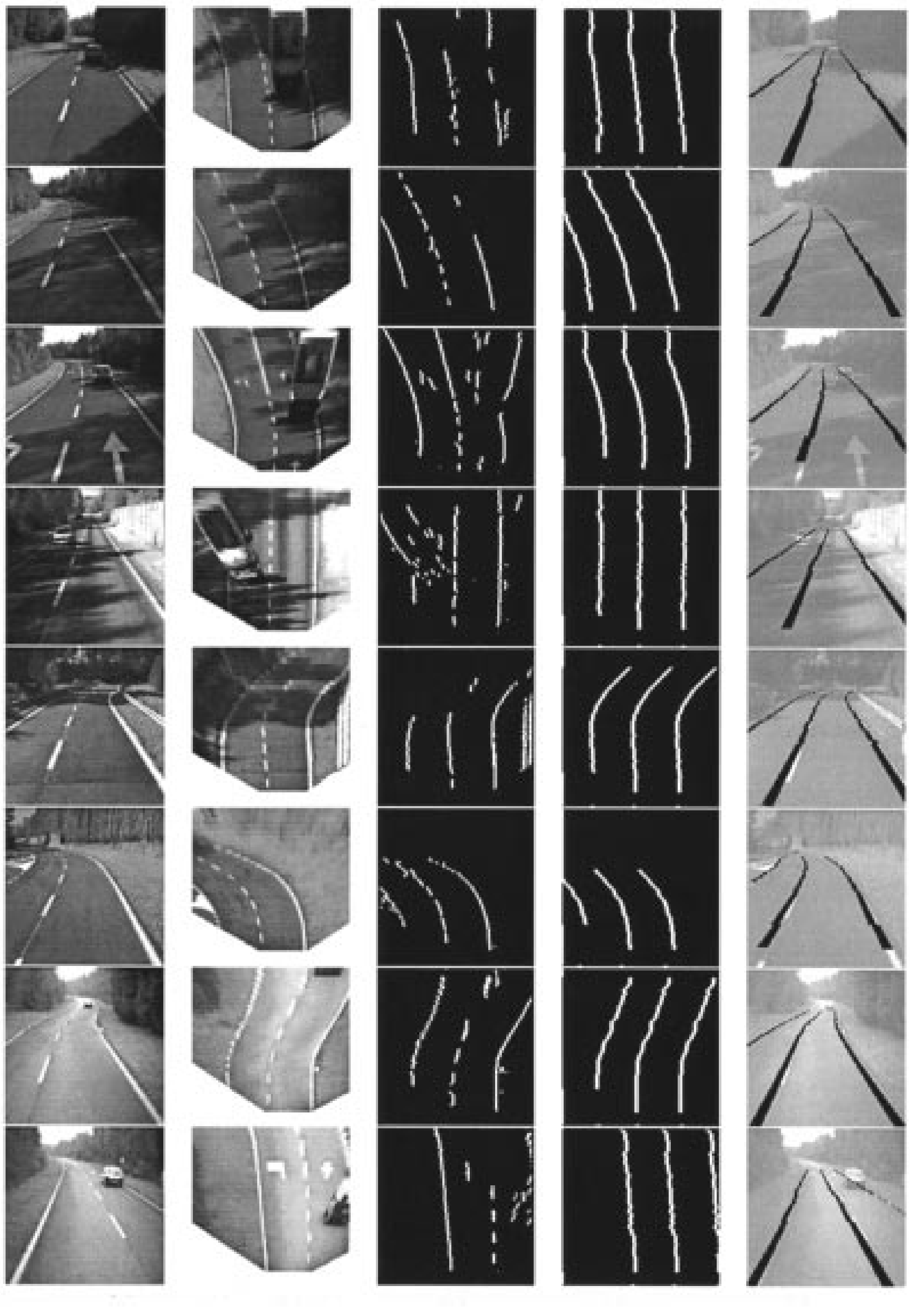


Fig. 13 Results of lane detection. (a) Original. (b) Remapped. (c) Low-level result. (d) Reconstructed. (e) Superimposed onto original.

Subsequently, all pairs (c_i, w_i) with $W - (W/4) < w_i < W + (W/4)$ are considered: the image is scanned line by line from its bottom (where it is more probable to be able to detect the center of the road) to its top, and the longest chain of road centers is built, exploiting the image vertical correlation. The values w_i corresponding to the centers c_i of the above chain are then used to reconstruct the road geometry. Fig. 12(a) shows the final result starting from the binary image shown in Fig. 9(e). Moreover, for displaying purposes, the perspective effect can be reintroduced using the

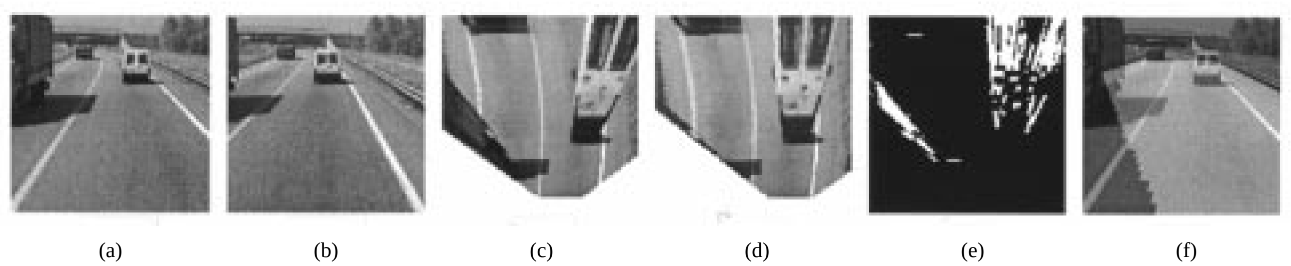


Fig. 14. Real situation with two obstacles. (a) Left view. (b) Right view. (c) Remapped left view. (d) Remapped right view. (e) Thresholded and filtered difference between remapped views. (f) In light gray, the road area visible from both cameras.

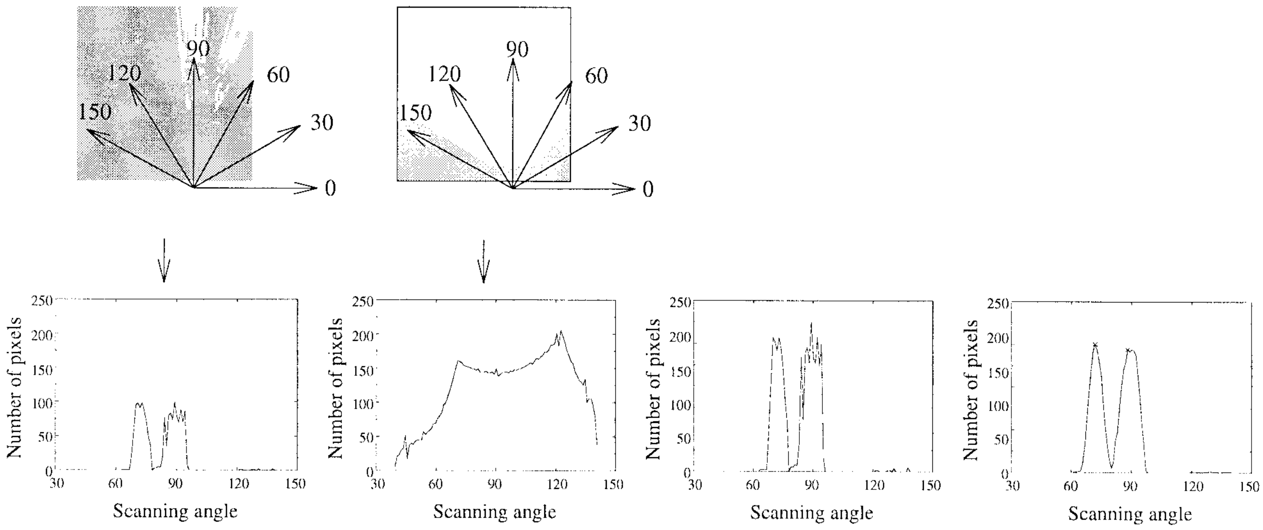


Fig. 15. Steps involved in the computation of the polar histogram.

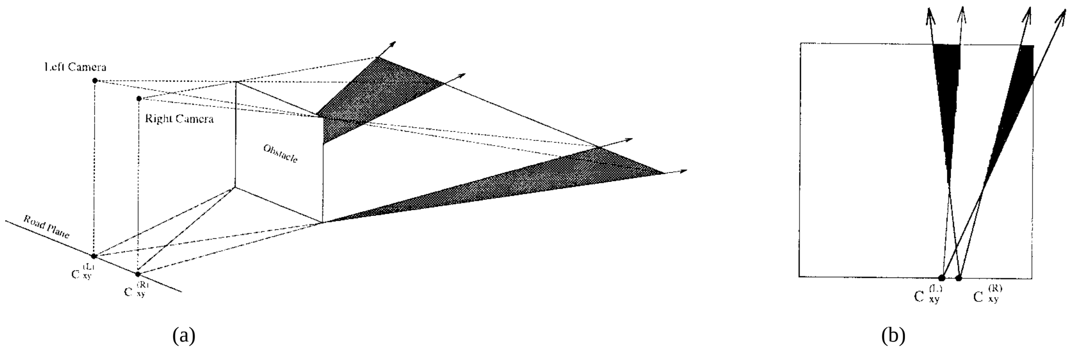


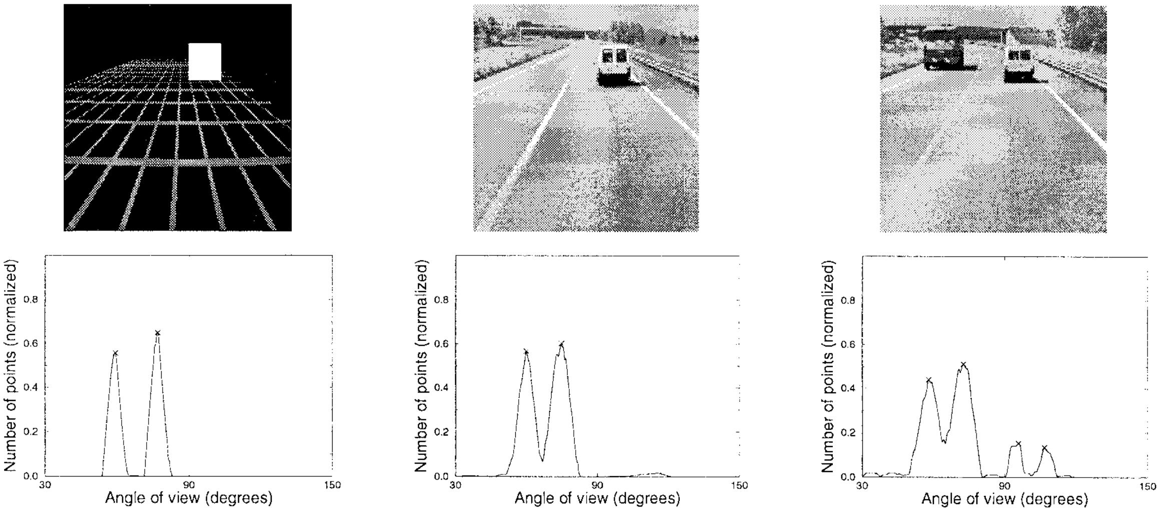
Fig. 16. Relationships between obstacle and cameras position. (a) Perspective view. (b) Road plane view.

dual transform of the IPM [see (1)]. Fig. 12(b) shows the representation in the $\mathcal{I}$ space of Fig. 12(a), while Fig. 12(c) presents its final superimposition onto the original image.

Fig. 17. Correspondence between triangles and directions pointed out by peaks detected in the polar histogram.

B. Obstacle Detection

As shown in Section III, the stereo IPM technique can produce a difference image in which ideal square obstacles are transformed into two triangles. The obstacle detection process is based on the localization of pairs of these triangles, which is derived from a quantitative measure of shape and position of the two corresponding triangles. Unfortunately, due to the texture, the irregular shape, and the nonhomogeneous brightness of real obstacles, the detection and localization of the triangles become difficult. Nevertheless, in the difference image some clusters of pixels with a quasitriangular shape are



(a) (b) (c)

Fig. 18. Polar histograms. (a) Ideal situation. (b) Real situation with one obstacle. (c) Real situation with two obstacles.

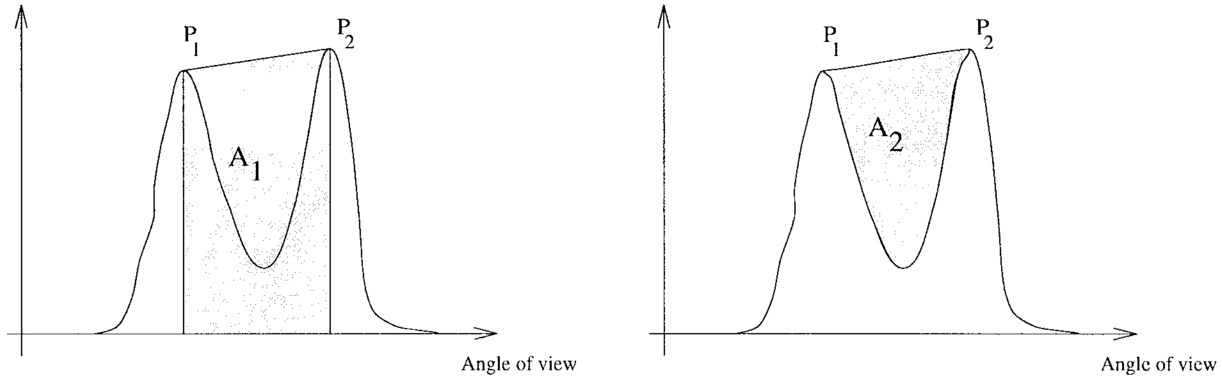


Fig. 19. If the ratio between areas A_1 and A_2 is greater than a threshold, the two peaks are joined.

anyway recognizable, even if they are not clearly disjointed. Moreover, in case two or more obstacles are present in the scene at the same time, more than two triangles appear in the difference image. A further problem is due to partially visible obstacles that produce a single triangle.

The low-level portion of the processing, detailed in Fig. 14, is thus reduced to the difference between the two remapped images, a threshold, and a morphological opening [27] aimed to the removal of small-sized details in the thresholded image.

1) Polar Histogram: A *polar histogram* is used for the detection of triangles: given a point F in the $z = 0$ plane of the $\mathcal{W}$ space (called *focus*), the polar histogram is computed scanning the difference image and counting the number of overthreshold pixels for every straight line originating from the focus F. The values of the polar histogram are then normalized using the polar histogram obtained scanning an image where all pixels are set (*reference image*). Furthermore, a lowpass filter is applied in order to decrease the influence of noise (see Fig. 15).

When F lays on the prolongation of a triangle edge, the polar histograms presents a sharp discontinuity. The localization of a triangle is thus performed, exploiting the following property: the prolongations of the edges of a triangle intersect both projections $C_{xy}^{(L)}$ and $C_{xy}^{(R)}$ of the two cameras onto the road plane, as shown in Fig. 16. Due to the small distance between $C_{xy}^{(L)}$ and $C_{xy}^{(R)}$, instead of computing two different polar histograms (having focus on $C_{xy}^{(L)}$ and $C_{xy}^{(R)}$), a single one is considered. The focus of the polar histogram is placed in the middle of $\overline{C_{xy}^{(L)}C_{xy}^{(R)}}$; in this case the polar histogram presents an appreciable peak corresponding to each triangle. Since the presence of an obstacle produces two disjoint triangles (corresponding to its edges) in the difference image, obstacle detection is reduced 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 (see Fig. 17). Peaks may have different characteristics, such as amplitude, sharpness, or width, depending on the obstacle distance, the angle of view, and the difference of brightness and texture between the background and the obstacle itself (see Fig. 18).

2) 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

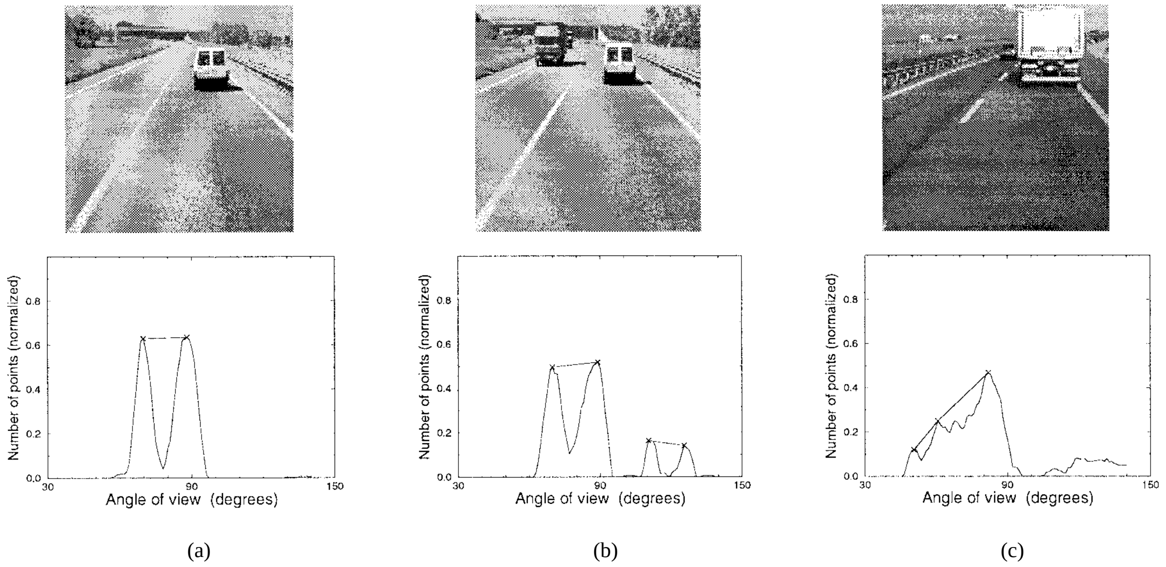


Fig. 20. Some examples of peaks join. (a) One obstacle. (b) Two obstacles. (c) A large obstacle.

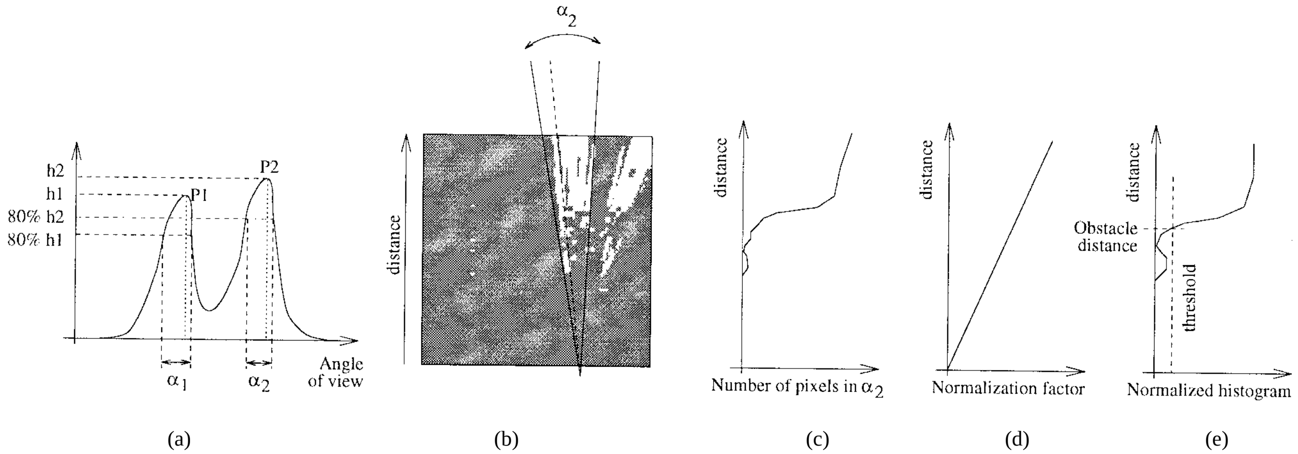


Fig. 21. Steps involved in the computation of radial histogram for peak P_2. (a) Polar histogram. (b) Binary difference image. (c) Radial histogram. (d) Normalization factor. (e) Normalized radial histogram.

situations allowed to determine a parameter embedding all of the above quantities. According to the notations of Fig. 19, 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 thus joined (see Fig. 20). Hence, the two peaks are left alone (not joined) when they are far apart or the valley is too deep. Obviously, a partially visible obstacle produces a single peak that cannot be joined to any other. The amplitude and width of peaks, as well as the interval between joined peaks, are used to determine the angle of view under which the whole obstacle is seen.

3) Estimation of Obstacle Distance: The difference image can also be used to estimate the obstacle distance. For each peak of the polar histogram, a *radial histogram* is computed scanning a specific sector of the difference image; the width α_i of the sector is determined as the width of the polar histogram peak in correspondence to the 80% of the peak maximum amplitude h_i, as shown in Fig. 21(a). The number of overthreshold pixels is computed and the result is normalized. The radial histogram is analyzed to detect the corners of triangles, which represent the contact points between obstacles and road plane, thus allowing the determination of the obstacle distance through a simple threshold (see Fig. 21).

The result is displayed with black markers superimposed on a brighter version of the left image; they encode both obstacles distance and width. Fig. 22 shows the results obtained in a number of different situations.

V. The Computing Architecture

Due to the specific field of application, the response time of the system is a major critical point, since it affects directly the maximum speed allowed for the vehicle; the choice of the computing architecture is, thus, a key design issue [10]. A few considerations on power consumption show that future trends in mobile computing are shifting toward a massively parallel

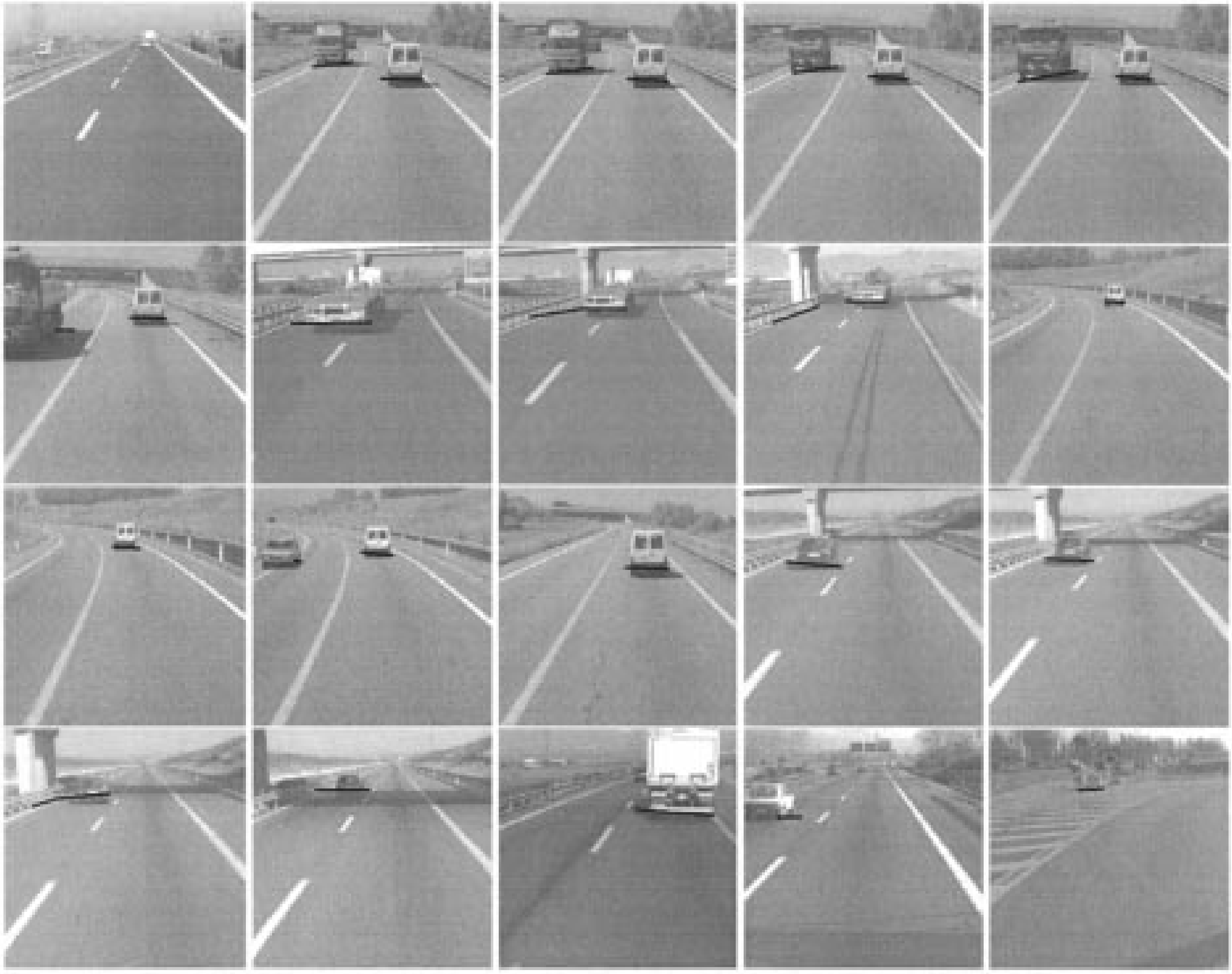


Fig. 22. Results of obstacle detection in different road conditions.

architecture composed of a large number of relatively slow-clocked processing elements (PE's). The power consumption of dynamic systems can be considered proportional to CfV^2, where C represents the capacitance of the circuit, f is the clock frequency, and V is the voltage swing. Power can be saved in the following three different ways [23], minimizing $C, f,$ and V respectively:

1) using a greater level of VLSI integration, thus reducing the capacitance C;
2) trading computer speed (with a lower clock frequency f) for lower power consumption (already implemented on many portable PC's);
3) reducing the supply voltage V_{DD}.

In recent years, IC supply voltage has been reduced from 5–3.3 V, and even less for the internal core. Unfortunately, there is a speed penalty to pay for this reduction: for a CMOS gate [54], the device delay T_d (following a first order approximation) is proportional to $V_{DD}/(V_{DD} - V_T)^2$, which shows that the reduction of V_{DD} determines a quasilinear increment (until the device threshold value V_T) of the circuit delay T_d. On the other hand, the reduction of V_{DD} determines a quadratic reduction of power consumption. Thus, for power saving reasons, it is desirable to operate at the lowest possible speed, but, in order to maintain the overall system performance, compensation for these increased delays is required [14], [16].

The reduction of power consumption, while maintaining computational power, can be reached by the use of low-cost SIMD computer architectures, formed by a large number of extremely simple and relatively slow-clocked PE's. These systems, using slower device speeds, provide an effective mechanism to trade power consumption for silicon area, while maintaining the computational power unchanged. Following are the four major drawbacks of this approach.

1) A solution based on hardware replication increases the silicon area, and thus it is not suited for extremely area-constrained designs.
2) Parallelism must be accompanied by extra-routing, requiring extra-power; this issue must be carefully considered and optimized.
3) The use of parallel computer architectures involves the redesign of the algorithms with a different computational model.
4) Since the number of processing units must be high, if the system has size constraints the PE's must be extremely simple, performing only simple basic operations.

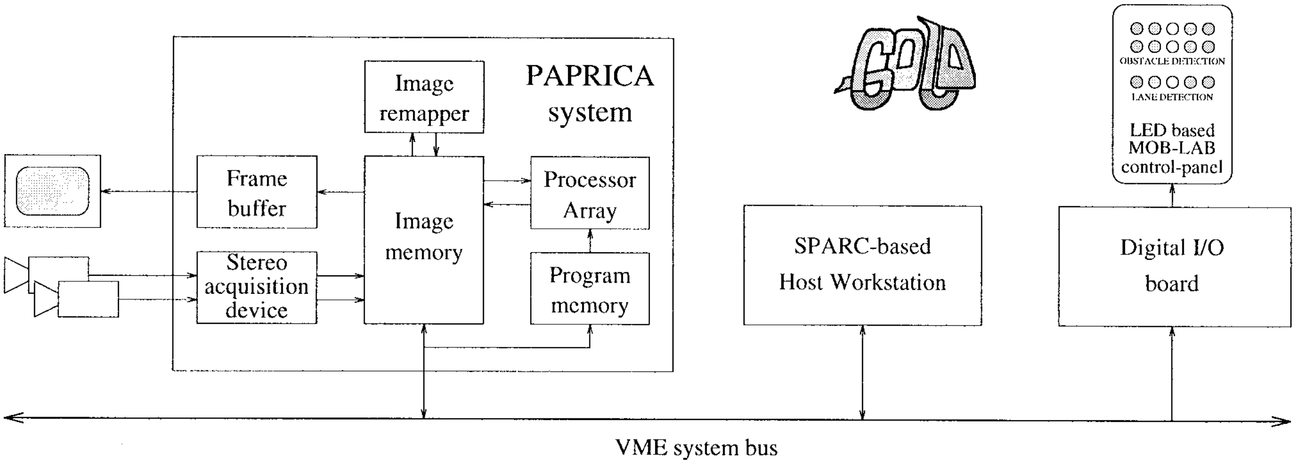


Fig. 23. Logical organization of GOLD system architecture.

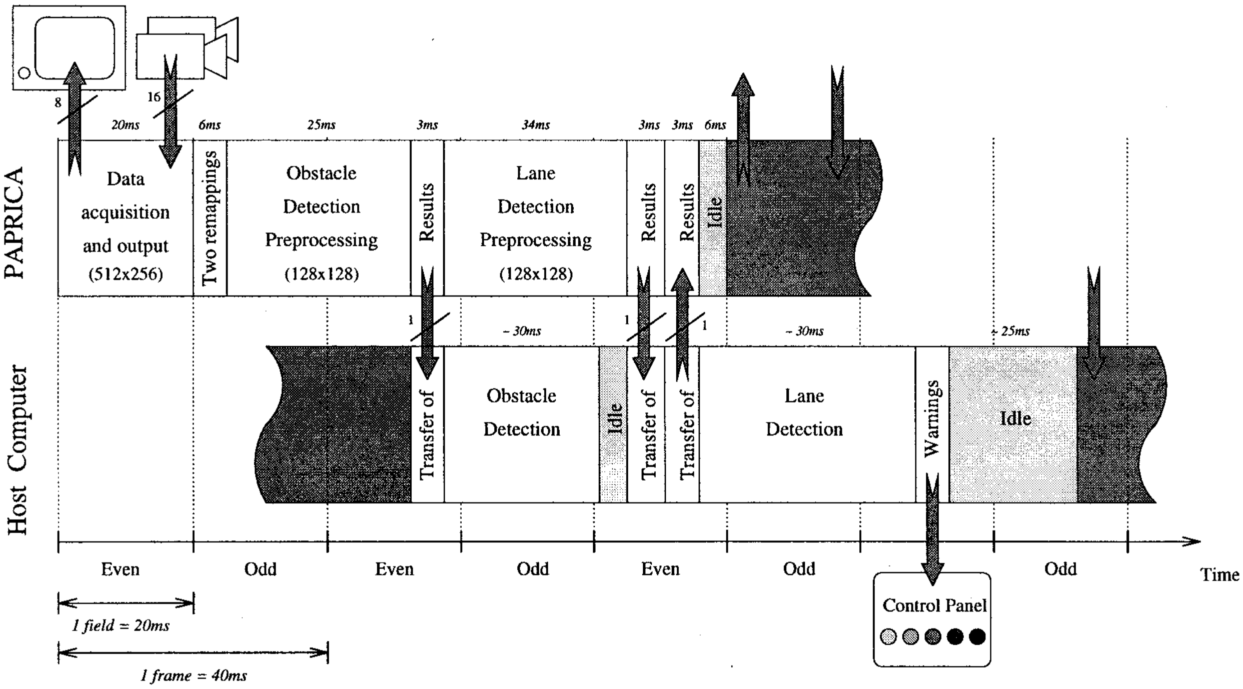


Fig. 24. Timings of the whole system.

More generally, the features that allow the integration and the efficient use of a computer system on board of a moving vehicle are 1) low production cost, 2) low operative cost, and 3) small physical size.

The *parallel processor for image checking and analysis* (PAPRICA) system [8], [9], a low-cost special-purpose massively parallel architecture composed of 256 PE's working in SIMD fashion, has been explicitly developed in cooperation with the Polytechnic Institute of Turin, Italy, following the above requirements. The PAPRICA coprocessor is integrated on a single VME board (6U) connected to a SPARC-based host workstation. It comprises the following five major functional parts:

- *program memory,* storing up to 256 000 instructions;
- *image memory* (up to 8 Mbytes);
- *processor array,* 16 × 16 square matrix of 1-b PE's each with full eight-neighbors connectivity [24], [40], [47], and an internal 64-b memory;
- *camera interface,* able to acquire stereo images and to display temporary results on an external monitor at video rate (25 frames/s or 50 fields/s);
- *control unit,* managing the activity of the whole system.

Generally in image processing applications, the number of PE's is far smaller than the number of image pixels, thus requiring a sort of virtualization of the PA. This is done utilizing an external image memory [40], [47], [51]; the processor array (PA) is loaded from the image memory with a subwindow of the data set; then the computation is performed until a special instruction is reached; finally, the results are stored back again into the external memory. These steps are

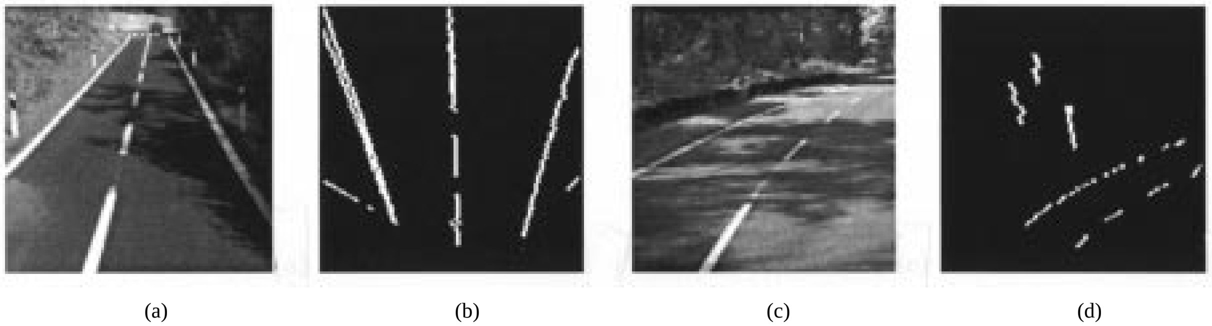


Fig. 25. Situations in which lane detection fails: (a) the road is not flat, thus producing (b) an irregular thresholded remapped image. (c) One of the road markings is not visible, thus producing (d) an incomplete thresholded remapped image.

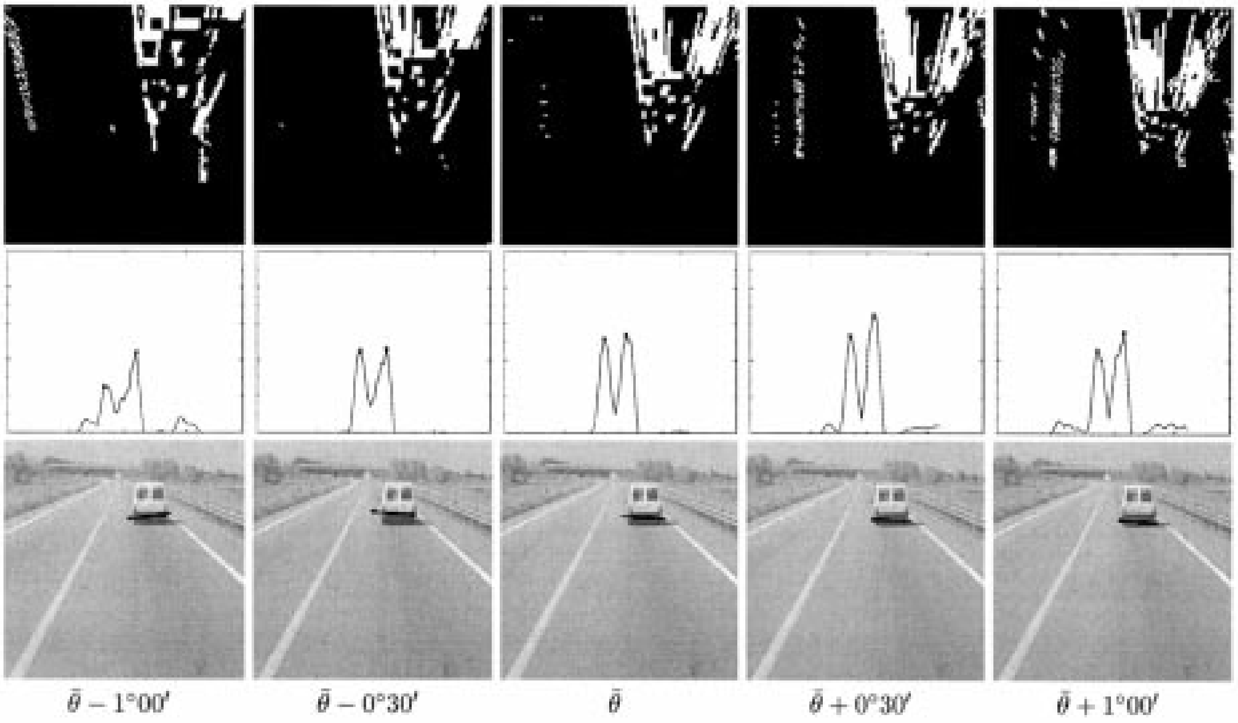


Fig. 26. Obstacle detection changing the inclination parameter: thresholded difference image, polar histogram, visual output of the processing, and value of the inclination parameter.

iterated until all the subwindows have been processed. Then the first subwindow is reloaded again into the PA and the computation is resumed.

The low-level portion of both lane detection and obstacle detection has been implemented on the PAPRICA system, while the medium-level part of the processing has been implemented on PAPRICA host computer. A block diagram depicting the logical organization of the computing engine is shown in Fig. 23.

A number of features are integrated into the PAPRICA system for the efficient implementation of the discussed driving assistance functions, as follows: 1) the morphological processing paradigm, which is fundamental for low-level processing; 2) hardware support for pyramidal vision tasks; 3) direct integration of an image acquisition device; and 4) hardware support for a fast image remapping.

1) Morphological Processing: The PAPRICA instruction set comprises two kinds of operations: graphical and logical. Graphical operators derive from mathematical morphology [27], [53], a bit-map approach to image processing based on set theory. For each pixel, the input to the graphical operator comes from one bit-plane of the pixel itself and from the *same* bit-plane of its eight surrounding neighbors: the output value depends on the particular morphological operator. The result of the graphical operator is then either stored in a destination bit-plane or used as the first operand of the following logical operation. Sources to the logical operator are the result of the graphical operator (the central pixel itself, if no operation was performed) and the value of a second bit-plane of the same pixel.

2) Support for Pyramidal Vision: Pyramidal architectures have shown several advantages in the field of image

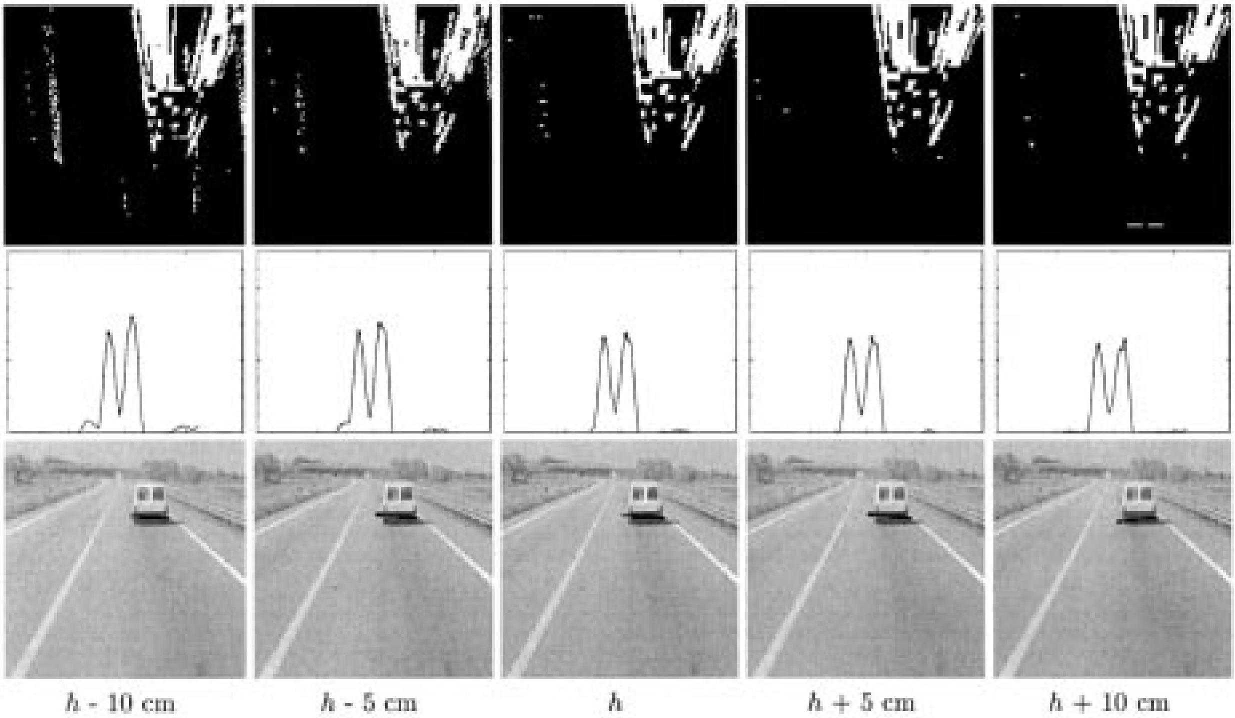


Fig. 27. Obstacle detection changing the height parameter: thresholded difference image, polar histogram, visual output of the processing, and value of the height parameter.

processing [12], [48], [55]. One major drawback of such architectures is that they are not easily scalable beyond a certain size, because interconnections among processors have an intrinsic 3-D structure that cannot be easily mapped onto a 2-D silicon surface. Thanks to the specific processor virtualization mechanism [4], PAPRICA allows to implement multiresolution algorithms with no dedicated hardware interconnections.

3) Camera Interface: The image acquisition device is integrated on board: it allows both to grab and store directly into PAPRICA image memory pairs of grey-tone (8 b/pixel) stereo images and to display the contents of the image memory on a monitor at video rate in two formats: 512 × 512 pixel at 25 Hz, corresponding to full frames, and 512 × 256 pixel at 50 Hz, corresponding to single fields (even or odd).

4) Data Remapping Support: A communication is defined to be *global* when the distance between the source and the destination coordinates are different for every pixel. With a 2-D mesh interconnection topology, it is impossible to perform a large number of global communications efficiently. The PAPRICA system includes a specific hardware device that allows the global movement of the image pixels without any constraints on their final destination. An *address* image is loaded into the image memory and treated as a look-up table; each pixel contains a pointer to a memory location where the source value for that pixel resides. A dedicated serial device, implemented in the same FPGA integrating PAPRICA controller, scans the address image and creates the new remapped image, in whose pixel values are determined by an indirect addressing. The remapping process takes three 50 ns clock cycles per pixel, giving a total of about 3 ms to generate a 128 × 128 remapped image. This device plays a basic role for the implementation of the GOLD system functionalities.

VI. Performance Analysis

Since the GOLD system is composed of two independent computational engines (the PAPRICA system, running the low-level processing, and its host computer, running the medium-level processing), it can work in pipelined. Therefore, the timing of the whole system is determined by the slowest process instead of the sum of the two. As shown in Fig. 24, the lane detection and obstacle detection tasks are divided into the following categories.

1) Data Acquisition and Output: A pair of grey-level stereo images of size 512 × 256 pixels are acquired simultaneously and written directly into PAPRICA image memory. At the same time, the result of previous computations are displayed on an on-board monitor to generate a visual feedback to the driver.

2) Remapping: The two acquired 512 × 256 pixel images are remapped into two 128 × 128 pixel images using two different look up tables in 6 ms.

3) Obstacle Detection Preprocessing: The difference between the two remapped images is computed and thresholded; a morphological filter is applied to the resulting image in order to remove noisy features. The result is a single binary 128 × 128 pixel image. This phase, again managed by PAPRICA system, takes 25 ms; the result is then transferred (in 3 ms) to the host computer.

4) Lane Detection Preprocessing: Also in this case, the result is a binary 128 × 128 pixel image. This phase, executed

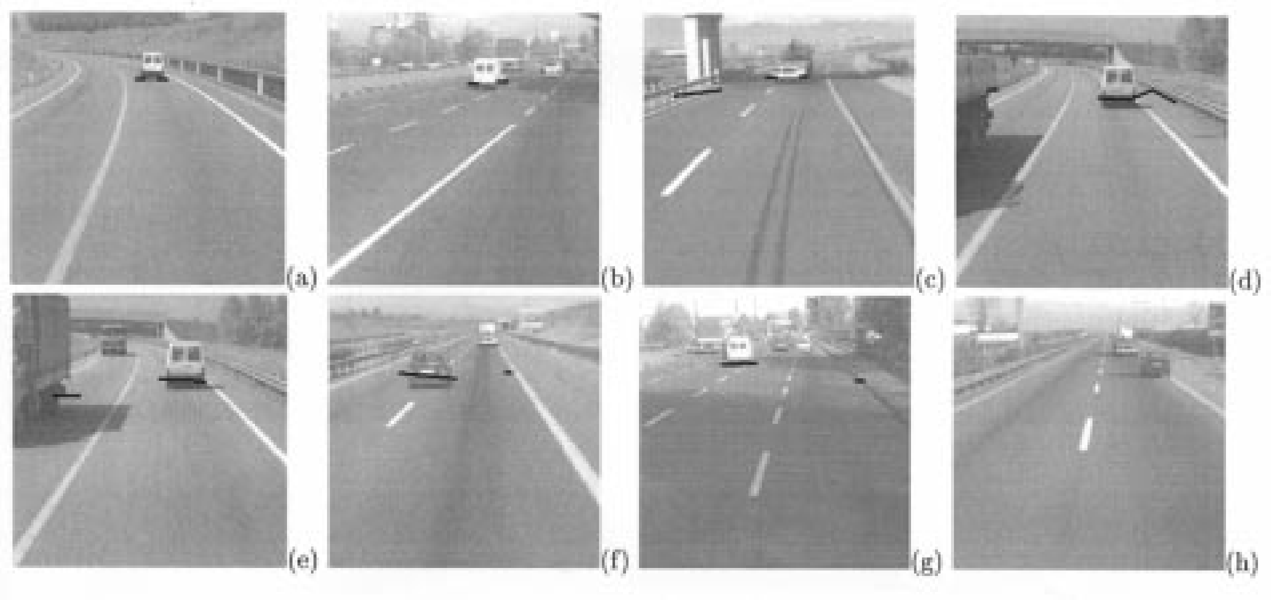


Fig. 28. Situations in which obstacle detection is critical.

on PAPRICA, takes 34 ms; the result (a 128 × 128 binary image) is transferred (in 3 ms) to the host computer.

5) Obstacle Detection: Since this computation is data-dependent, only an estimate of the average processing time required by the host computer can be made, ranging from 20–30 ms. Then the result is transferred to PAPRICA memory to be displayed during the acquisition of the following frame.

6) Lane Detection: The computation is again data-dependent; the average processing time is about 30 ms.

7) Warnings: The last phase of the whole computational cycle is the displaying of results on the control panel, issuing warnings to the driver. The time required for this phase is negligible with respect to the whole processing.

Since, for our purposes, a 512 × 256 image resolution has demonstrated to be sufficient, the PAPRICA frame-grabber is used in single-field acquisition mode; thus, images become available at the rate of 50 per second (using standard cameras at 25 Hz). Therefore, the whole processing is divided into 20 ms time slots. For this reason, once its processing is over, the PAPRICA system remains idle until the beginning of the following time slot.

As shown in Fig. 24, the whole processing (lane and obstacles detection) requires five time slots (100 ms);[2] the GOLD system works at a rate of 10 Hz.

VII. Discussion

In this work, a system (hardware and software) for lane and obstacle detection has been presented, satisfying the hard real-time constraints imposed by the automotive field. The whole system was tested with images acquired by the stereo vision system installed on-board of the experimental vehicle MOB-LAB. It was driven for more than 3000 km, along extra-urban roads and freeways, under different traffic and illumination conditions, at speeds up to 80 km/h. GOLD is now being tested also on ARGO.

Obviously, when the initial assumptions are not met, namely when the *road is not flat* [Fig. 25(a) and (b)] or when *road markings are not visible* [Fig. 25(c) and (d)], lane detection cannot produce valid results.

On the other hand, since obstacle detection uses stereo vision, the quality of results is tightly coupled also with the calibration of the vision system. Nevertheless, since the final target of obstacle detection is the determination of the free space in front of the vehicle and not the complete 3-D reconstruction of the world, camera calibration becomes less critical. For this reason, even if the movements of the vehicle modify some of the calibration parameters (camera height h and inclination $\overline{\theta}$ with respect of the road plane), a dynamic recalibration of the system is not required[3]. For comparison purposes, the ranging values for cameras height $(h \pm 10 \text{ cm})$ and inclination $(\overline{\theta} \pm 1^\circ)$ larger than the ones estimated in [33] have been considered. Figs. 26 and 27 show the results of obstacle detection emulating the changes of cameras parameters caused by vehicle movements: due to the robustness of the approach based on polar histogram, the obstacle is always detected even if the difference images are noisy.

The major critical problems of obstacle detection were found when:

- the obstacle is too far from the cameras (generally, this happens in the range 45–50 m), thus the polar histogram presents only small and isolated peaks that can be hardly joined [Fig. 28(a)–(c)]; however, when the obstacle distance is in the range 5–45 m, this problem has never been detected;

[2] The result displayed on the control panel have a latency of 150 ms.

[3] Should the recalibration be required, it could be done during the idle time of the host computer: the IPM recalibration consists of loading a new *address image* (look-up table) into the PAPRICA remapping device.

- the guard-rail is close to the obstacle—and thus a single large obstacle—is detected [Fig. 28(d)];
- an obstacle is partially visible, and thus only one of its edges can be detected [Fig. 28(d) and (e)];
- some noisy peaks in the polar histogram are not filtered out, and thus they are considered as small obstacles [Fig. 28(f) and (g)];
- the detection of far obstacles sometimes fails when their brightnesses is similar to the brightnesses of the road [Fig. 28(h)].

The confidence in the detection of obstacles is obviously dependent on their size, distance, and shape, and specific parameters are used to tune the system sensitivity.

1) Obstacle Height: The obstacle height determines the amplitude of peaks in the polar histogram. The bandwidth of the lowpass filter (LPF) applied to the polar histogram is the parameter used as threshold to discard small peaks that could be caused by either noise or short obstacles: the smaller the bandwidth, the lower the influence of noise (caused by incorrect camera calibration or vehicle movements), but the larger the minimum height of detectable obstacles [in Fig. 28(c) and (d), the guard-rail is detected even if it is not as tall as vehicles].

2) Obstacle Width: In the polar histogram, two or more peaks generated by the same obstacle are joined when the ratio A_1/A_2 (see Section IV-B2) is greater than a threshold. In the worst case (when the peaks are separated by deep valleys), the smaller this threshold, the larger the width of the wider correctly detectable object, but the higher the probability that peaks not generated by the same obstacle are joined [see Fig. 28(d)].

3) Obstacle Distance: The farther the obstacle, the smaller the portion of triangles detectable in the difference image, and thus the lower the amplitude of peaks in the polar histogram; nevertheless, for sufficiently high obstacles (e.g., vehicles at about 50 m far from the cameras), the main problem is not the detection of peaks, but their joining, as shown in Fig. 28(a)–(c).

4) Obstacle Shape: The algorithm was designed to detect obstacles with quasivertical edges; objects with nonvertical edges (e.g. pyramidal objects) generate twisted triangles that are hardly detected by the analysis of the polar histogram.

Moreover, since only a portion of the incoming image is analyzed [see Fig. 14(f)], a different setup of the stereo cameras could both increase the resolution of the remapped image and remove the problems due to the framing of a portion of a bright sky.

Also, the intercamera spacing is a key parameter: the greater the distance between cameras, the stronger the disparities in the the remapped images due to the presence of an obstacle. Nevertheless the intercamera spacing is bounded by the vehicle physical structure, thus the cameras were installed at the maximum allowed distance, without running other experiments with larger intercamera spacings. Unfortunately, a too large separation leads to a higher sensitivity to vehicle movements, in particular, rolling.

During the tests, the system demonstrated to be robust and reliable: vehicles were *always* detected and only in few cases (i.e., on rough or paved roads) vehicle movements become so high that the processing of noisy remapped images led to the erroneous detection of false small sized obstacles.

On the other hand, thanks to the remapping process, lane markings were located even in presence of shadows or other artifacts on the road surface; anyway, although it is hard to devise a method to evaluate the percentage of successful lane detection, some unofficial tests showed that the system detects the correct position of the lane in about 95% of the situations considered; however, this result does not take into account an exhaustive set of road conditions.

The above-mentioned problems are mainly due to the independent processing of single images, without taking into account the high temporal correlation among subsequent frames in a sequence. Considering an operational vehicle speed of 100 km/h and the MOB-LAB calibration setup, the vertical shift between two subsequent remapped images corresponding to two frames acquired with a temporal shift of 100 ms is only 7 pixels. This high correlation allows to average in time the results of the processing, thus reducing the problems of the incomplete detection of obstacles explained above. Moreover, a temporal averaging should decrease also the influence of the noise introduced by vehicle movements, since it can be modeled as a high-frequency sinusoid.

An extension to the GOLD system that is able to exploit temporal correlations and to perform a deeper datafusion between the two functionalities of lane detection and obstacle detection is currently under test [2] on ARGO. As shown in Fig. 24, the idle time of the host computer (about 25 ms for each cycle) allows in fact to perform a more complex processing without affecting the overall system performance.

ACKNOWLEDGMENT

The authors express their gratitude to E. Dickmanns for his outstanding suggestions, to F. Gregoretti, L. Reyneri, C. Sansoé, and R. Passerone of the Polytechnic Institute of Torino, for the enthusiastic joint development of the PAPRICA system, and to G. Quaglia and all the friends from IEN Galileo Ferraris, Torino, for their help during the tests on MOB-LAB. The authors also acknowledge the significative contribution of all the students who were involved in this project, in particular, A. Fascioli. Finally, the authors are also in debt to G. Conte and G. Adorni for their support in this research.

REFERENCES

[1] M. Bertozzi, A. Broggi, and S. Castelluccio, “A real-time oriented system for vehicle detection,” *J. Syst. Architecture*, vol 43, pp. 317–325, Mar. 1997.

[2] M. Bertozzi, A. Broggi, and A. Fascioli, “Obstacle and lane detection on ARGO autonomous vehicle,” in *Proc. IEEE Intelligent Transportation Systems Conf.’97*, Boston, MA.

[3] S. Beucher and M. Bilodeau, “Road segmentation and obstacle detection by a fast watershed transform,” in *Proc. IEEE Intelligent Vehicles ’94*, Paris, France, Oct. 1994, pp. 296–301

[4] A. Broggi, “Performance optimization on low-cost cellular array processors,” in *Proc. MPCS–IEEE Int. Conf. Massively Parallel Computing Systems*, Ischia, Italy, May 1994, pp. 334–338.

[5] ——, "Parallel and local feature extraction: A real-time approach to road boundary detection," *IEEE Trans. Image Processing*, vol. 4, pp. 217–223, Feb. 1995.

[6] ——, "Robust real-time lane and road detection in critical shadow conditions," in *Proc. IEEE Int. Symp. Comput. Vis.*, Coral Gables, FL, Nov. 19–21, 1995,

[7] A. Broggi and S. Bertè, "Vision-based road detection in automotive systems: A real-time expectation-driven approach," *J. Artif. Intell. Res.*, vol. 3, pp. 325–348, Dec. 1995.

[8] A. Broggi, *et al.*, "Design and implementation of the PAPRICA parallel architecture," *J. VLSI Signal Process.*, to be published.

[9] —— "The evolution of the PAPRICA system," *Integrated Comput.-Aided Eng. J.—Spec. Issue Massively Parallel Comput.*, vol. 3, no. 4, 1996.

[10] A. Broggi and F. Gregoretti, Eds., "Editorial: Special issue on special-purpose architectures for real-time imaging," *Real-Time Imag. J.*, vol. 2, pp. 329–330, Dec. 1996.

[11] P. J. Burt, P. Anandan, and K. J. Hanna, "An electronic front end processor for active vision, in *Proc. SPIE Conf. Intell. Robot.*, Nov. 1992.

[12] V. Cantoni and M. Ferretti, *Pyramidal Architectures for Computer Vision*. London, U.K.: Plenum, 1993.

[13] S. Carlsson and J. O. Eklundh, "Object detection using model-based prediction and motion parallax," in *Proc. Europ. Conf. Computer Vision*, Antibes, France, Apr. 1990, vol. 427, pp. 297–306.

[14] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.

[15] R. M. H. Cheng, J. W. Xiao, and S. LeQuoc, "Neuromorphic controller for AGV steering, in *Proc. IEEE Int. Conf. Robotics and Automation*, 1992, pp. 2057–2062.

[16] B. Courtois, "CAD and testing of IC's and systems: Where are we going?," Tech. Rep., TIMA & CMP, Jan. 1993.

[17] J. Crisman and C. Thorpe, "SCARF: A color vision system that tracks roads and intersections," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 49–58, Feb. 1993.

[18] E. D. Dickmans and B. D. Mysliwetz, "Recursive 3-D road and relative ego-state recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 199–213, May 1992.

[19] F. Diebolt, "Reconaissance des marquages routièrs," Ph.D. dissertation, L'Univ. Louis Pasteur de Strasbourg, Strasbourg, France, Dec. 1996.

[20] W. Enkelmann, "Obstacle detection by evaluation of optical flow field from image sequences," in *Proc. Europ. Conf. Comput. Vis.*, Antibes, France, Apr. 1990, vol. 427, pp. 134–138.

[21] W. Enkelmann, G. Struck, and J. Geisler, "ROMA: A system for model-based analysis of road markings," in I. Masaky, Ed., *Proc. IEEE Intelligent Vehicle '95 Conf.*, Detroit, MI, pp. 356–360.

[22] A. Fascioli, "Localizzazione di ostacoli mediante elaborazione di immaini stereoscopiche." Master's thesis, Univ. Studi di Parma, Parma, Italy, 1995.

[23] G. H. Forman and J. Zahorjan, "The challenge of mobile computing," *Computer*, vol. 27, pp. 38–47, Apr. 1994.

[24] T. Fountain, *Processor Arrays: Architectures and Applications*. New York: Academic, 1987.

[25] A. Giachetti, M. Campani, R. Sanni, and A. Succi, "The recovery of optical flow for intelligent cruise control," in *Proc. IEEE Intelligent Vehicles '94*, Paris, France, pp. 91–95.

[26] V. Graefe and K.-D. Kuhnert, "Vision-based autonomous road vehicles," in I. Masaki, Ed., *Vision-Based Vehicle Guidance*. New York: Springer-Verlag, 1991, pp. 1–29.

[27] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathemtical morphology," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 532–550, 1987.

[28] T. M. Jochem, D. A. Pomerleau, and C. E. Thorpe, "MANIAC: A next generation neurally based autonomous road follower," in *Proc. Int. Conf. Intelligent Autonomous Systems: IAS-3*, Pittsburgh, PA, Feb. 1993.

[29] K. Kluge, "Extracting road curvature and orientation from image edge points without perceptual grouping into features," in *Proc. IEEE Intelligent Vehicles '94*, pp. 109–114.

[30] K. Kluge and G. Johnson, "Statistical characterization of the visual characteristics of painted lane markings," in *Proc. IEEE Intelligent Vehicles '95*, Detroit, MI, pp. 488–493.

[31] K. Kluge and S. Lakshmanan, "A deformable template approach to lane detection," in *Proc. IEEE Intelligent Vehicles '95*, Detroit, MI, pp. 54–59.

[32] K. Kluge and C. Thorpe, "Intersection detection in the YARF road following system," in *Proc. IEEE Int. Conf. Intelligent Autonomous Systems*, Tokyo, Japan, 1993, pp. 145–155.

[33] D. Koller, J. Malik, Q.-T. Luong, and J. Weber, "An integrated stereo-based approach to automatic vehicle guidance," in *Proc. Fifth ICCV*, Boston, MA, 1995, pp. 12–20.

[34] J. Kosecka, R. Bajcsy, and M. Mintz, "Control of visually guided behaviors," in *Real-Time Computer Vision*, C. Brown and D. Terzopoulos, Eds. Cambridge, U.K.: Cambridge Univ. Press, 1994.

[35] W. Kruger, W. Enkelmann, and S. Rossle, "Real-time estimation and tracking of optical flow vectors for obstacle detection," in *Proc. IEEE Intelligent Vehicles '95*, Detroit, MI, pp. 304–309.

[36] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biol. Cybern.*, vol. 64, pp. 177–185, 1991.

[37] L. Matthies, "Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation," *Int. J. Comput. Vis.*, vol.8, pp.71–91,1992.

[38] M. Meng and A. C. Kak, "Fast vision-guided mobile robot navigation using neural networks," in *Proc. IEEE Conf. Systems, Man, and Cybernetics '92*, pp. 111–116.

[39] L. E. Mettala and O. Firschein, "Reconnaissance, surveillance and target acquisition research for unmanned ground vehicle program," in *Proc. Int. Conf. Intelligent Autonomous Systems*, 1993, pp. 257–279.

[40] NCR Corporation, Dayton, OH, *Geometric Arithmetic Parallel Processor*, 1984.

[41] W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics*. New York: McGraw-Hill, 1981.

[42] M. Ohzora *et al.*, "Video-rate image processing system for an autonomous personal vehicle system," in *Proc. IAPR Workshop on Machine Vision and Applications*, Tokyo, Japan, 1990, pp. 389–392.

[43] D. A. Pomerleau, "Neural network based autonomous navigation," in C. E. Thorpe, Ed., *Vision and Navigation: The Carnegie Mellon Navlab*. Boston, MA: Kluwer, 1990, pp. 83–93.

[44] ——, *Neural Network Perception for Mobile Robot Guidance*. Boston, MA: Kluwer, 1993.

[45] ——, "RALPH: Rapidly adapting lateral position handler," in *Proc. IEEE Intelligent Vehicles '95*, Detroit, MI, Sept. 25–26, 1995, pp. 506–511.

[46] D. Raviv and M. Herman, "A new approach to vision and control for road following," in *Proc. Conf. Computer Vision and Pattern Recognition*, June 1991, pp. 217–225.

[47] S. Reddaway, "DAP—A distributed array processor," in *1st Ann. Symp. Computer Architectures*, 1973, pp. 61–65.

[48] A. Rosenfeld, *Multiresolution Image Processing and Analysis*. New York: Springer-Verlag, 1984.

[49] B. Ross, "A practical stereo vision system," in *Proc. Int. Conf. Computer Vision and Pattern Recognition*, Seattle, WA, June 21–23, 1993, pp. 148–153.

[50] Y. Ruichek and J.-G. Postaire, "Real-time neural vision for obstacle detection using linear cameras," in *Proc. IEEE Intelligent Vehicles '95*, Detroit, MI, pp. 524–529.

[51] L. A. Schmitt and S. S. Wilson, "The AIS-5000 parallel processor," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 320–330, May 1988.

[52] H. Schneidermann and J. S. Albus, "Progress and prospects for vision-based automatic driving," in *Proc. 1993 Ann. Meet. IVHS AMERICA*, pp. 569–575.

[53] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.

[54] M. Shoji, *CMOS Digital Circuit Technology*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[55] S. L. Tanimoto, T. J. Ligocki, and R. Ling, "A prototype pyramid machine for hierarchical cellular logic," in *Parallel Computer Vision*. New York: Academic, 1987.

[56] F. Thomanek, E. D. Dickmanns, and D. Dickmanns, "Multiple object recognition and scene interpretation for autonomous road vehicle guidance," in *Proc. IEEE Intelligent Vehicles '94*, Paris, France, pp. 231–236.

[57] Y. Wan, F. Caestaing, and J. Burie, "A new edge detector for obstacle detection with a linear stereo vision system," in *Proc. IEEE Intelligent Vehicles '95*, Detroit, MI, pp. 130–135.

[58] J. Weber, D. Koller, Q.-T. Luong, and J. Malik, "New results in stereo-based automatic vehicle guidance," in *Proc. IEEE Intelligent Vehicles '95*, Detroit, MI, pp. 530–535.

[59] X. Yu, S. Beucher, and M. Bilodeau, "Road trackin, lane segmentation and obstacle recognition by mathematical morphology," in *Proc. IEEE Intelligent Vehicles '92*, pp. 166–170.

[60] Y. Zheng, *et al.*, "SWITCHER: A stereo algorithm for ground plane obstacle detection," *Image Vis. Comput.*, vol. 8, pp. 57–62, 1990.

[61] T. Zielke, M. Brauckmann, and W. von Seelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP: Image Understand.*, vol. 58, pp. 177–190, 1993.

Massimo Bertozzi (SM'95) received the B.S. degree in electronic engineering from the University of Parma, Parma, Italy. He is currently a Ph.D. student at the Department of Information Engineering, University of Parma. His research interests are mainly focused on the application of image processing to vehicle guidance and the optimization of machine code.

Mr. Bertozzi is the Chairman of the local IEEE Student Branch.

Alberto Broggi (S'90–A'96) received the Dr.Ing. degree in electronic engineering in 1990 and the Ph.D. degree in information technology in 1994 from the University of Parma, Parma, Italy.

He has been a Researcher at the Department of Information Engineering, University of Parma, since 1994. His research interests include real-time computer vision algorithms for the navigation of unmanned vehicles and the development of low-cost computer systems to be used on autonomous robots. He is the main investigator of the University's ARGO project, involved in the design, development, and testing of the ARGO autonomous prototype vehicle.

Dr. Broggi is the author of more than 70 referred publications in international Journals, book chapters, and conference proceedings. He is the Editor of the Newsletter and member of the Executive Committee of the IEEE Technical Committee on Complexity in Computing, and member of the Editorial Board of *Real-Time Imaging Journal*. He has guest edited a number of special issues of international journals on machine vision, including an issue of *IEEE Expert Magazine* on vision-based driving assistance in vehicles of the future. He has been invited to organize several minitracks and special sessions in IEEE-sponsored international conferences. He has served on the Program Committees and as tutorial chairman of many major conferences.