Real-Time Lane and Obstacle Detection on the GOLD System*

Massimo Bertozzi and Alberto Broggi
Dipartimento di Ingegneria dell'Informazione
Università di Parma, I-43100 Parma, Italy
e-mail: {bertozzi,broggi}@CE.UniPR.IT
Phone: +39-521-905707 Fax: +39-521-905723

Abstract

This paper describes the GOLD (Generic Obstacle and Lane Detection) system, a stereo vision-based hardware and software architecture developed to increment road safety of moving vehicles: 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). It has been implemented on the PAPRICA system and works at a rate of 10 Hz.

1 Introduction

The GOLD system addresses both lane detection and obstacle detection at the same time: lane detection is based on a pattern-matching technique which 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 3D world reconstruction [8].

Both functionalities share a the same underlying approach (*inverse perspective mapping* [9]), which is based on the assumption of a flat road. Such a technique is based on a transform that, given a model of the road in front of the vehicle (e.g. flat road) remaps both images into a new domain (road domain), in which the following processings are extremely simplified. The transform causes the removal of the perspective effect, thus each pixel represents the same portion of the road. In this way lane markers have a constant width within the whole image, thus simplifying their detection. The use of image warping on stereo images allows to detect any object that raises out from the road plane. Moreover, since both GOLD functionalities are based on the processing of images remapped into the same domain (ground plane), the fusion of the result of the two independent processings is straightforward.

This work is organized as follows: section 2 presents the basics of the approach used to remove the perspective effect from a monocular image and its application to the processing of stereo images. Section 3 describes how lane detection is performed while section 4 describes obstacles detection. Section 5 describes the hardware architecture and section 6 shows the implementation and performance. Section 7 ends the paper with a brief discussion of the results.

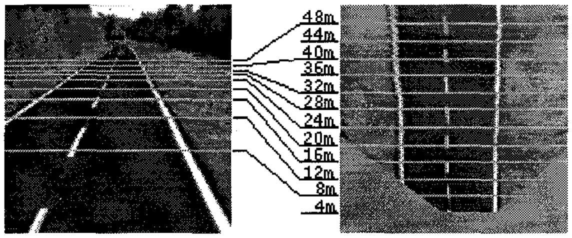

Figure 1: The original and the remapped images

2 The Underlying Approach

The *Inverse Perspective Mapping* (IPM) is a well-established technique [2, 10, 12, 14, 15] that allows to remove the perspective effect when the acquisition parameters (camera position, orientation, optics,...) are completely known and when a knowledge about road is given, such as a *flat road hypothesis*. The procedure aimed to remove the perspective effect resamples the incoming image, remapping each pixel toward a different position and producing a new 2-dimensional array of pixels. The resulting image represents a top view of the road region in front of the vehicle, as it were observed from a significant height. The image acquired by a camera and its remapping are shown in figure 1, where also a relationship between the two coordinate systems is given.

Obviously, since in the *remapped image* lane markings have a constant width, lane detection can be performed even in critical conditions, such as presence of shadows [3]. On the other hand, the remapping of *stereo images* can deliver information about the presence of elevated obstacles or, in general, of a *non-flat* portion of the road.

*This work was partially supported by CNR Progetto Finalizzato Trasporti under contracts 93.01813.PF74 and 93.04759.ST74.

0-7803-3652-6/96/$5.00 © IEEE

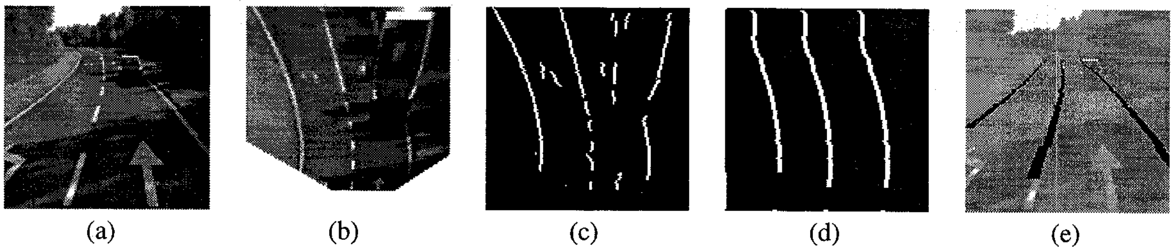

Figure 2: Results of Lane Detection: a) original; b) remapped; c) tresholded; d) reconstructed; e) superimposed onto original

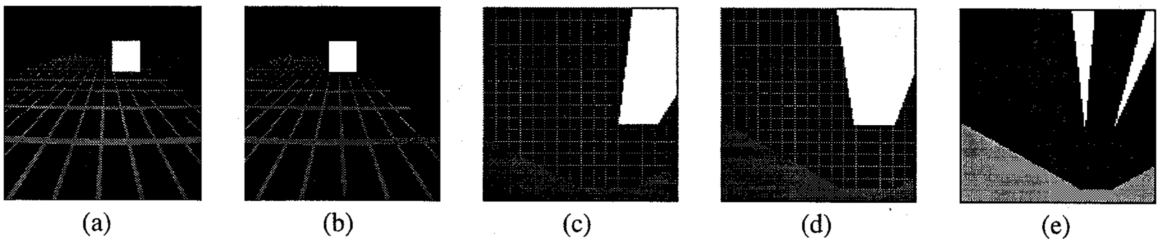

Figure 3: An ideal white object on a dark gridded background: a) left view; b) right view; c) remapped left view; d) remapped right view; e) difference between (c) and (d) showing in light grey the area not seen by both cameras

3 Lane Detection

Lane detection is performed assuming that a road marking in the road plane (i.e. in the remapped image) is represented by a quasi-vertical 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. So the first phase of road markings detection is thus based on a linewise determination of horizontal black-white-black transitions. A horizontal brightness gradient-based filter is used to compute a thresholded image, figure 2.b and 2.c show the remapped and the thresholded image correspondent to figure 2.a.

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 all its non-zero pixels are considered in pairs. Two non-zero pixels can represent three different road configurations, depending on the meaning that they may have: left edge, center line, right edge.

Each one of the 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:

$$\begin{cases} 0 \le c_i \le N \\ w_i < \frac{N}{3} \\ c_i - w_i \le \frac{3}{4}N \\ c_i + w_i \ge \frac{N}{4} \end{cases}$$

where N is the image horizontal size.

An histogram is built considering all the values assumed by the w_i parameter for each line of the image, where the peak of the histogram corresponds to the more frequent value of the lane width. In order to allow a non-fixed road geometry (and also the handling of curves) the histogram is low-pass filtered; finally its maximum value W is detected. Subsequently, all pairs (c_i, w_i) with:

$$W - \frac{W}{4} < w_i < W + \frac{W}{4}$$

are considered: the image is scanned line by line from its bottom (where it is more probable to be able to detect successfully 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. 2.d shows the final result starting from the original image shown in figure 2.a. Moreover, for displaying purposes, the perspective effect can be reintroduced using the dual transform of the IPM: figure 2.e presents its final superimposition onto the original image.

4 Obstacle Detection

Since a flat road is assumed, 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, two 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 pairs of pixels having the same image coordinates in the two remapped images represent the same point in the road plane.

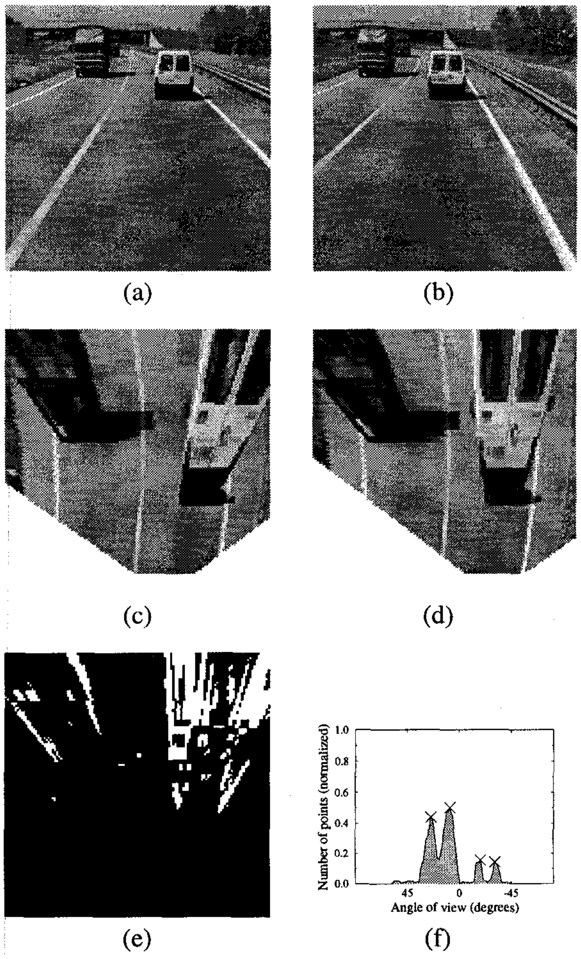

Figure 4: Obstacle detection in case of two vehicles with different shape and color in front of the stereo system: (a) left image; (b) right image; (c) left remapped image; (d) right remapped image; (e) difference image; (f) polar histogram showing four maxima corresponding to two objects

Thus the flat road hypothesis can be verified computing the difference between the two remapped images: anything raising out from the road is detected if the difference image presents sufficiently large clusters of non-zero pixels.

In addition it is easily demonstrable that the vertical edges of a generic obstacle are mapped into two straight lines intersecting in the projection of the camera onto the road plane. Due to the different coordinates of the projection of the two cameras, in an *ideal case* the difference between the two remapped images presents two triangles (see figure 3).

Unfortunately, in a *real case*, due to the texture, the irregular shape, and the non-homogeneous brightness of obstacles, the detection and localization of the triangles become difficult. Nevertheless in the computation of the difference image some clusters of pixels with a quasi-triangular shape are anyway recognizable.

The low-level portion of the processing is thus reduced to the difference between the two remapped images, a threshold, and a simple morphological opening [7] aimed to the removal of small-sized details in the thresholded image.

A *polar histogram* is used for the detection of the triangles: given a point *F* (called *focus*) in the road plane 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*. If *F* lays on the prolongation of a triangle edge, the polar histograms presents a sharp discontinuity. Thus if *F* is placed in the middle point (as in figure 5) between the projections of the two cameras, the polar histogram presents an appreciable peak corresponding to each triangle. Figures 4.a and 4.b show the left and right views of two obstacles, figures 4.c and 4.d show the left and right remapped stereo images, while the difference image is represented in figure 4.e, while figure 4.f shows the correspondent polar histogram. Moreover two or more peaks can be joined according to different criteria, such as similar amplitude, closeness, or sharpness. 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.

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 goal of the analysis of radial histogram is to detect the corners of triangles, which represent the contact points between obstacles and road plane, thus allowing the determination of the obstacle distance. Further details can be found in [5].

5 The Hardware System

The low-level processing of images is handled by a special purpose full-custom massively parallel system: PAPRICA, developed in cooperation with Politecnico di Torino, Italy [4]. PAPRICA is a massively parallel SIMD computer architecture devoted to the processing of 2D data structures; it is composed of a square matrix of 256 single-bit Processing Elements (PEs) disposed on the nodes of a 2D mesh, each one with full 8-neighbors connectivity [11, 6, 13]. In the current prototype the Processor Array is composed of an array of 4 × 4 full custom ICs, each of them containing a sub-array of 4 × 4 PEs. Each PE has 64 internal registers. The PAPRICA low-level processor, integrated on a single VME board (6U), comprises 6 major functional parts:

1. the Program Memory, storing up to 256k instructions;
2. the Image Memory, up to 8 MBytes of 35 ns RAM;
3. the Processor Array, whose instruction cycle time is 250 ns;
4. the Frame Grabber device, able to grab (and store directly into PAPRICA Image Memory) 512×512

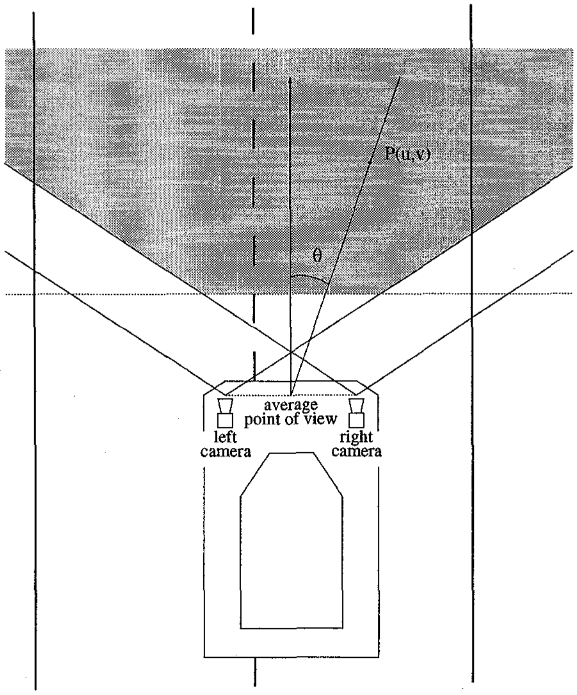

Figure 5: The determination of the polar histogram: the value of the pixels of the difference image are added for θ ranging in the interval $\left[-\frac{\pi}{2},+\frac{\pi}{2}\right]$, the grey part represents the common view area of both cameras

8 bit/pixel grey-tone images at video rate (25 frames/s);

5. the Image Remapper, used to perform the IPM;
6. the Control Unit, managing the activities of the whole system.

After the low-level portion of the processing, the resulting images are analyzed by PAPRICA front-end (a SPARC-based system).

6 Implementation and Performance

Since the GOLD system is composed by two autonomous computational engines (the PAPRICA system, performing the low-level processing, and its host computer, performing the medium-level processing), it can work as a pipelined structure. Therefore the timing of the whole system is determined by the slowest processing instead of the sum of the two. As shown in fig. 6 the lane detection and obstacle detection tasks are divided in:

Data acquisition and output: a pair of grey-level stereo images of size 512 × 256 pixel are acquired simultaneously and written directly into PAPRICA image memory. At the same time the result of previous computations can be optionally displayed on an on-board monitor to generate a visual feedback to the driver.

Remapping: the acquired images are remapped in 128 × 128 pixel images. Even if PAPRICA is able to remap a pair of stereo images at the same time (requiring 3 ms for a 128 × 128 pixel remapping process), in this case different look up tables are needed for left and right images, so this step is performed twice, thus requiring 6 ms. This step is required both for lane detection and for obstacle detection.

Obstacle detection preprocessing: the difference between the two remapped images is computed and thresholded; the result is a single binary 128 × 128 pixel image. This phase, again performed by PAPRICA system, takes 25 ms, then the result is transferred (taking 3 ms) to the host computer.

Lane detection preprocessing: also in this case the result is a binary 128 × 128 pixel image. This phase, executed on PAPRICA, takes 34 ms; the result (a 128 × 128 binary image) is transferred (in 3 ms) to the host computer.

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 to 30 ms. Then the result is transferred to PAPRICA memory to be displayed during the acquisition of the following frame.

Lane detection: the computation is again data-dependent; the processing time is about 30 ms.

Warnings: the last phase of the whole computational cycle is the displaying of results on a specific 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, PAPRICA frame-grabber is used in single-field acquisition mode, and thus images become available at the rate of 50 per second (using standard European 25 Hz cameras). Therefore the whole processing is divided into 20 ms time slots. For this reason, once its processing is over, PAPRICA system remains idle until the beginning of the following time slot.

As shown in fig. 6, the whole processing (lane and obstacles detection) requires five time slots (100 ms)[1], thus the GOLD system works at a rate of 10 Hz.

7 Conclusion

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 main innovative contribution of this work is the use of the IPM technique to simplify both low and medium level processing steps.

The system was tested on MOB-LAB 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 has proven to be reliable in a number of different situations (see figure 7 and 8) and to be more robust with respect to noise (such as shadows) than previous approaches based on monocular vision [1].

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

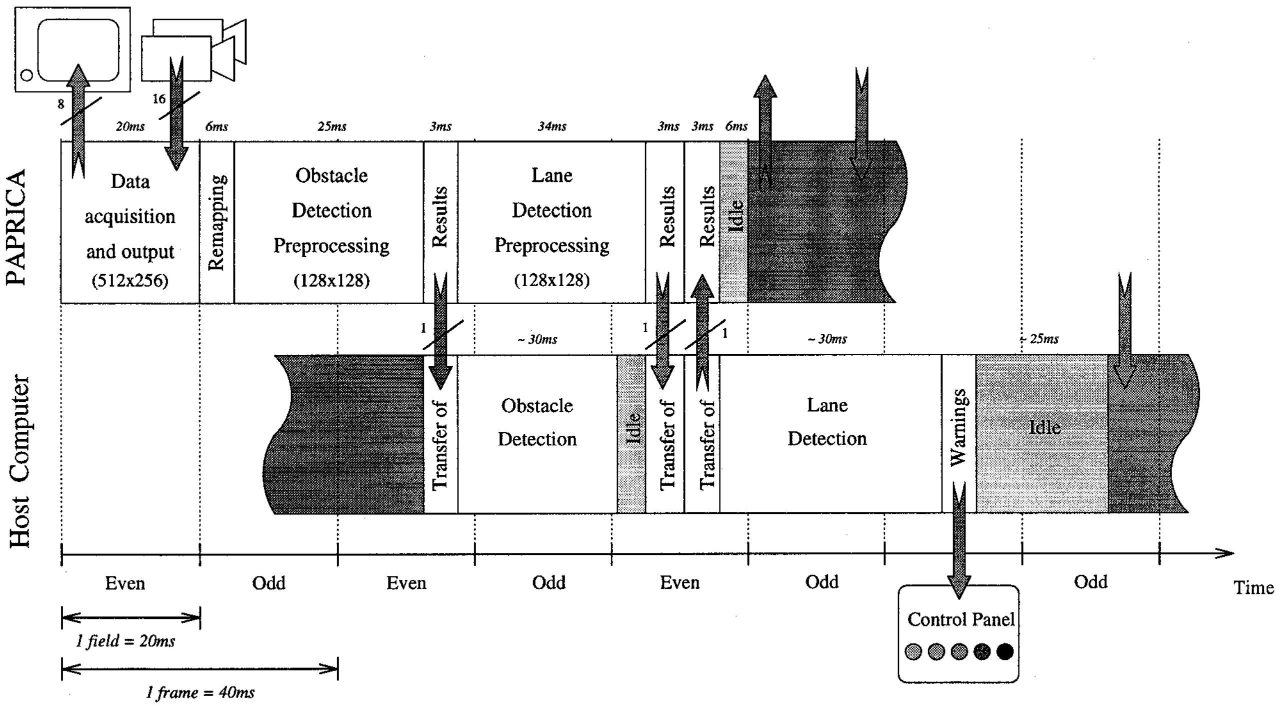

Figure 6: Timings of the whole system

Moreover it is currently under development an extension to the GOLD system able to exploit temporal correlations and to perform a deeper data-fusion between the two functionalities of lane detection and obstacle detection; as shown in fig. 6 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.

References

[1] M. Bertozzi, A. Broggi, and S. Castelluccio. A Real-Time Oriented System for Vehicle Detection. *Journal of Systems Architecture*, 1996. In press.

[2] A. Broggi. A Massively Parallel Approach to Real-Time Vision-Based Road Markings Detection. In I. Masaky, editor, *Proceedings IEEE Intelligent Vehicles'95*, pages 84–89, Detroit, September 25-26 1995. IEEE Computer Society.

[3] A. Broggi. Robust Real-Time Lane and Road Detection in Critical Shadow Conditions. In *Proceedings IEEE International Symposium on Computer Vision*, Coral Gables, Florida, November 19-21 1995. IEEE Computer Society.

[4] A. Broggi, G. Conte, F. Gregoretti, C. Sansoè, and L. M. Reyneri. The Evolution of the PAPRICA System. *Integrated Computer-Aided Engineering Journal - Special Issue on Massively Parallel Computing*, 4(1), 1996. In press.

[5] A. Fascioli. Localizzazione di ostacoli mediante elaborazione di immagini stereoscopiche. Master's thesis, Università degli Studi di Parma - Facoltà di Ingegneria, 1995.

[6] T. Fountain. *Processor Arrays: Architectures and applications*. Academic-Press, London, 1987.

[7] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image Analysis Using Mathematical Morphology. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 9(4):532–550, 1987.

[8] D. Koller, J. Malik, Q.-T. Luong, and J. Weber. An integrated stereo-based approach to automatic vehicle guidance. In *Proceedings of the Fifth ICCV*, pages 12–20, Boston, 1995.

[9] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64:177–185, 1991.

[10] L. Matthies. Stereo vision for planetary rovers: stochastic modeling to near real-time implementation. *International Journal of Computer Vision*, 8:71–91, 1992.

[11] NCR Corporation, Dayton, Ohio. *Geometric Arithmetic Parallel Processor*, 1984.

[12] M. Ohzora, T. Ozaki, S. Sasaki, M. Yoshida, and Y. Hiratsuka. Video-rate image processing system for an autonomous personal vehicle system. In *Proceedings IAPR Workshop on Machine Vision and Applications*, pages 389–392, Tokyo, 1990.

[13] S. Reddaway. DAP - A Distributed Array Processor. In *1st Annual Symposium on Computer Architectures*, pages 61–65, Florida, 1973.

[14] D. Pomerleau. RALPH: Rapidly Adapting Lateral Position Handler. In I. Masaky, editor, *Proceedings IEEE Intelligent Vehicles'95*, pages 506–511, Detroit, September 25-26 1995. IEEE Computer Society.

[15] Y. Ruichek and J.-G. Postaire. Real-Time Neural Vision for Obstacle Detection Using Linear Cameras. In I. Masaky, editor, *Proceedings IEEE Intelligent Vehicles'95*, pages 524–529, Detroit, September 25-26 1995. IEEE Computer Society.

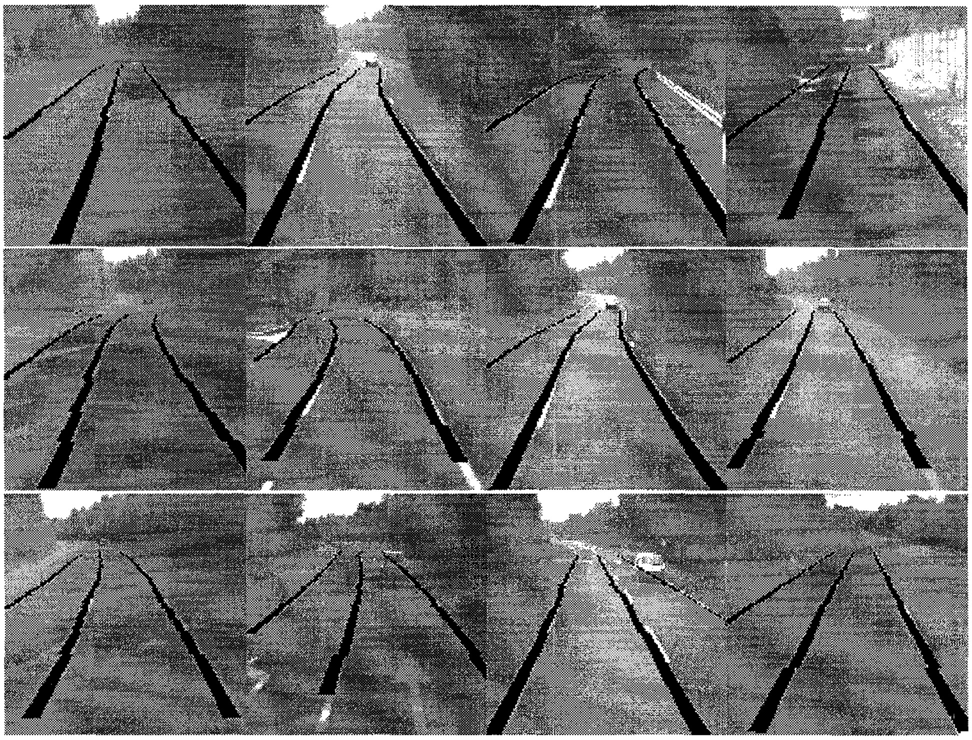

Figure 7: Results of Lane Detection with or without shadows and other painted signs

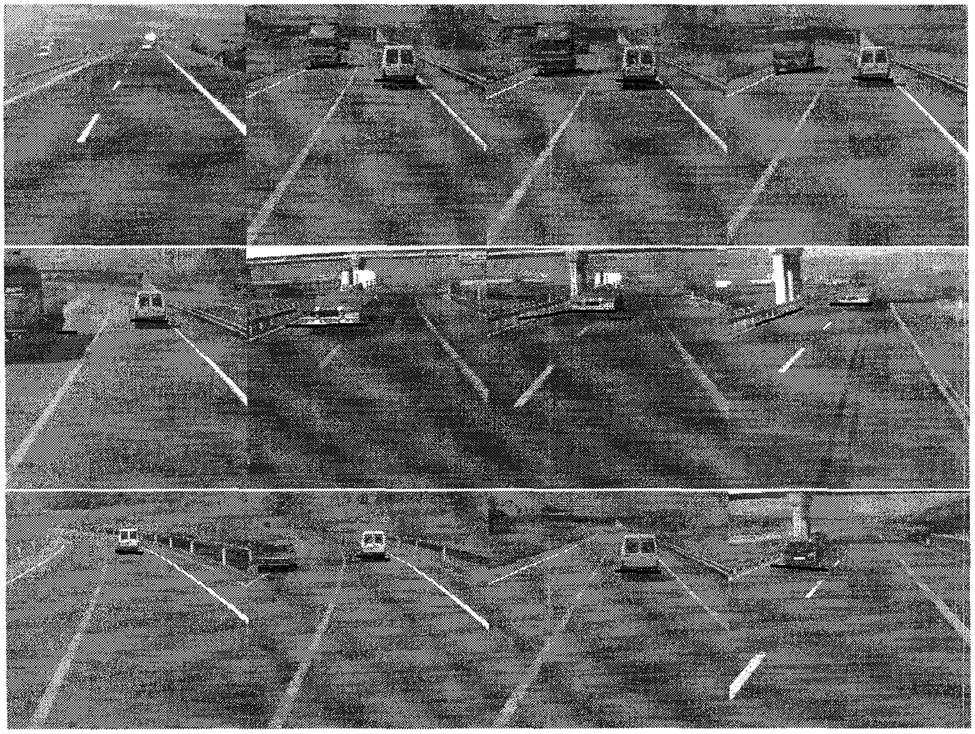

Figure 8: Results of Obstacle Detection in different road conditions, a black marker superimposed on a brighter version of the left image encodes both the distances and the width of the obstacles