A STEREO VISION SYSTEM
FOR REAL-TIME AUTOMOTIVE OBSTACLE DETECTION

Massimo Bertozzi, Alberto Broggi, Alessandra Fascioli

Dipartimento di Ingegneria dell'Informazione
Università di Parma, Italy
e-mail: {bertozzi,broggi,fascal}@CE.UniPR.IT

ABSTRACT

This work presents a system for obstacle detection in pair of images acquired by a stereo vision device installed on a moving vehicle. The whole system is structured in a pipeline of two different computational engines: a massively parallel architecture, PAPRICA, devoted to low-level image processing and a traditional serial architecture running medium-level tasks.

A geometrical transformation, based on the assumption of a flat road in front of the vehicle, is performed to remove the perspective effect from both images. The difference between the results is used for the detection of free-space in front of the vehicle, thus allowing to avoid the high computational tasks involved in *traditional* stereo vision approaches; the geometrical transformation is performed by a specific hardware device integrated in PAPRICA architecture.

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 demonstrated its robustness with respect to shadows and changing illumination conditions, different road textures, and vehicle movement.

1. INTRODUCTION

This work presents a stereo vision system (hardware and software) aimed to the detection of obstacles on the path of a mobile road vehicle.

Since *real time* and *low-cost* are the most important constraints in the automotive field, *traditional* stereo vision approaches, requiring extremely powerful computational engines, can not satisfy the low-cost requirements imposed by a widespread use. But if obstacle detection is reduced to the determination of the free-space in front of the vehicle (thus intending as "obstacle" anything raising out from the road surface), a complete 3D world reconstruction is no more required. Moreover a partial a-priori knowledge, such as the assumption of a flat road in front of the vehicle, allows the detection of correspondences in stereo images with a simpler check step [11, 8, 10]. The use of a special purpose massively parallel architecture that works in a pipeline fashion with a serial architecture allows to perform both the low-level and medium-level processing at the same time.

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 and how obstacles detection is performed. Section 3 describes the hardware architecture, while section 4 depicts the implementation details. Section 5 ends the paper with a discussion of the results.

2. THE APPROACH

2.1. The Inverse Perspective Mapping

The *Inverse Perspective Mapping* (IPM) is a well established technique [2, 11, 13, 4, 6] that allows to remove the perspective effect when the acquisition parameters (camera position, orientation, optics,...) are completely known and when the road in front of the vehicle is *flat*. The processing of *monocular* images obtained after the application of the IPM has already been successful in the determination of the road markings [3] even in critical shadow conditions [4]; on the other hand, the IPM process applied to *stereo pairs* can deliver information about the presence of elevated obstacles or, in general, of a *non-flat* portion of the road[1].

Assuming a flat road and no obstacles in front of the vehicle, the removal of the perspective effect produces a new image representing the texture of the road surface as it were observed from the top. Thus, the IPM process applied to both left and right images (acquired concurrently) produces two images representing the *same* texture. Any difference in these two images represents a deviation from the assumption of *flat* road, and thus identifies an obstacle.

2.2. Obstacles Detection

As mentioned above, the successive analysis of the *difference image* leads to the determination of possible obstacles: in the *difference image* obstacles are generally represented by two triangles (see figure 3.e), corresponding to the two edges of the obstacle seen from the two different angles of view of the stereo system. The medium-level processing, performed by the serial system, consists of scanning the *difference image* in order to produce a polar histogram (see figure 3.f),

THIS WORK WAS PARTIALLY SUPPORTED BY CNR PROGETTO FINALIZZATO TRASPORTI UNDER CONTRACTS 93.01813.PF74 AND 93.04759.ST74.

---

[1] Note that also road bumps and dips are here detected as obstacles

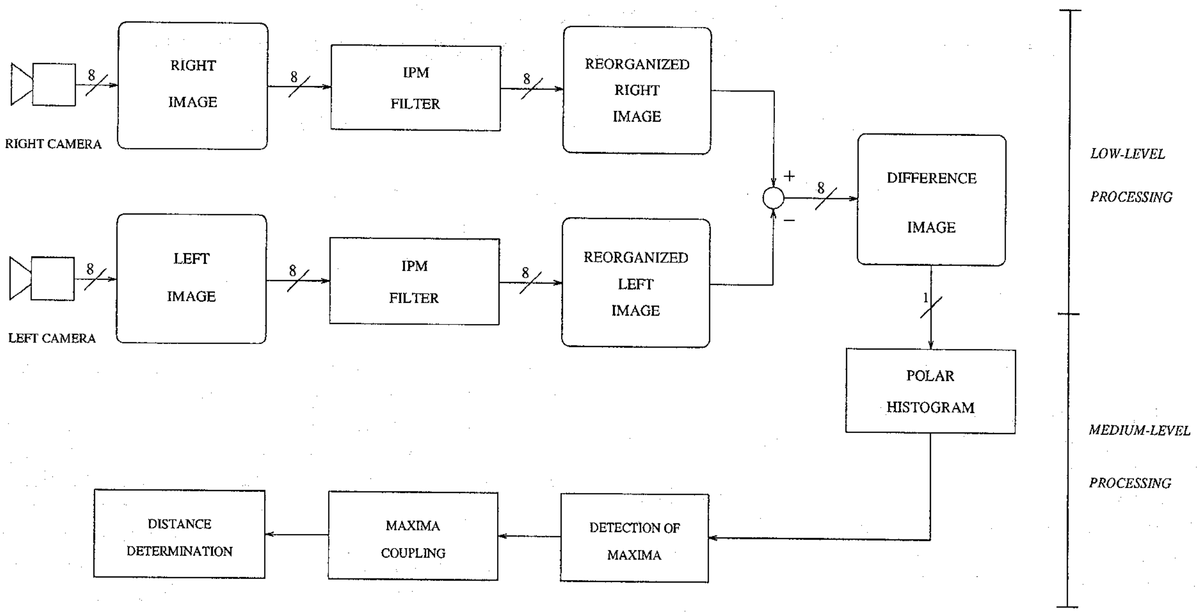

Figure 2: Block diagram of the algorithm

grouping all the pixels belonging to the same vertical edge of the obstacles. Figure 1 shows how the polar histogram is computed. The maxima of such a histogram are then grouped together (two adjacent maxima with similar characteristics, such as area or amplitude, generally represent a single obstacle) and the obstacle detected. The distance from the obstacle is determined by a further analysis of the *difference image* in the directions pointed out by the maxima determined at the previous step. A block diagram of the algorithm is sketched in figure 2.

3. HARDWARE SYSTEM

The low-level processing of images is handled by a special purpose full-custom massively parallel system: PAPRICA (PArallel PRocessor for Image Checking and Analysis), developed in cooperation with Politecnico di Torino, Italy [5]. 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 [9, 7, 12]. In the current prototype the Processor Array is composed of an array of 4×4 full custom ICs (1.5 μm CMOS, 45 mm^2, $\simeq$ 35000 transistors), 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 8 bit/pixel grey-tone images at video rate (25 frames/s);
5. the Image Remapper, used to perform the IPM;

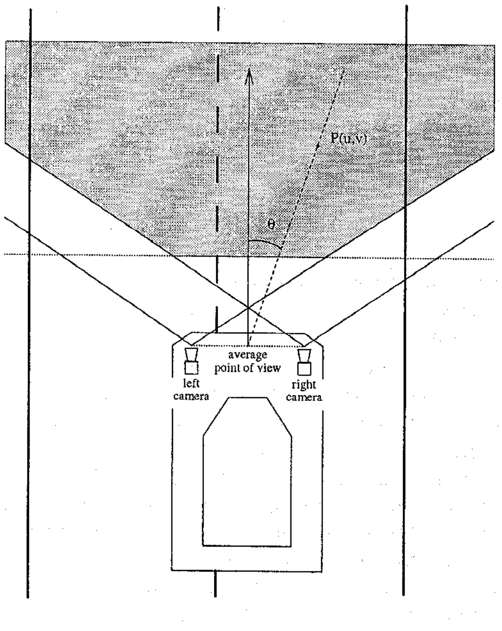

Figure 1: The determination of the polar histogram: the value of the pixels of the difference image are added for θ ranging in the interval $\left[-\frac{\pi}{2}, +\frac{\pi}{2}\right]$, the gray part represents the common view area of both cameras

6. the Control Unit, managing the activities of the whole system.

After the low-level portion of the processing, the resulting image is analyzed by PAPRICA front-end (a SPARC-based system) and the position of possible obstacles is determined.

4. IMPLEMENTATION

As mentioned above, the PAPRICA subsystem acquires pairs of stereo images at video rate (25 full frames/s or 50 image fields/s). The image reorganization is performed by a dedicated hardware module of the PAPRICA system (the Image Remapper): a 128 × 128 pixel image is reorganized in less than 3 ms. The original left and right images are shown in figures 3.a and 3.b, while the reorganized images are shown in figures 3.c and 3.d.

The successive processing, still performed in parallel by the PAPRICA system, is aimed to determine the disparities between the two reorganized images, with a neighborhood-based threshold with hysteresis (the resulting binary image is shown in figure 3.e). Figure 3.f shows the polar histogram which presents 2 maxima, corresponding to the 2 triangles in the *difference image*. The histogram, the detection of its maxima, the maxima grouping, and the determination of the obstacle distance from the camera are performed serially by PAPRICA front-end. The result (a black marker) is superimposed onto a brighter version of the original image and displayed onto PAPRICA frame buffer.

5. DISCUSSION

Since the 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 in pipeline. Therefore the timing of the whole system is determined by the slowest processing instead of the sum of the two. The whole process is 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.
- 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.
- 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.
- 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.

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, the whole obstacle detection requires up to 3 slots, thus working at a rate of 16 Hz.

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 (with none, one, or more obstacles with different shapes and colors, see the set of final images in figure 4) and to be more robust with respect to noise (such as shadows) than previous approaches based on monocular vision [1].

6. 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] S. Bohrer, T. Zielke, and V. Freiburg. An Integrated Obstacle Detection Framework for Intelligent Cruise Control on Motorways. In I. Masaky, editor, *Proceedings IEEE Intelligent Vehicles'95*, pages 276–281, Detroit, September 25-26 1995. IEEE Computer Society.

[3] 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.

[4] 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.

[5] A. Broggi, G. Conte, F. Gregoretti, C. Sansoè, and L. M. Reyneri. The PAPRICA Massively Parallel Processor. In *Proceedings MPCS - IEEE International Conference on Massively Parallel Computing Systems*, pages 16–30, Ischia, Italy, May 2-6 1994. IEEE Computer Society - EuroMicro.

[6] P. J. Burt, P. Anandan, and K. J. Hanna. An electronic front end processor for active vision. In *SPIE Conf on Intelligent Robotics*, November 1992.

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

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

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

[10] 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.

[11] 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.

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

[13] 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.

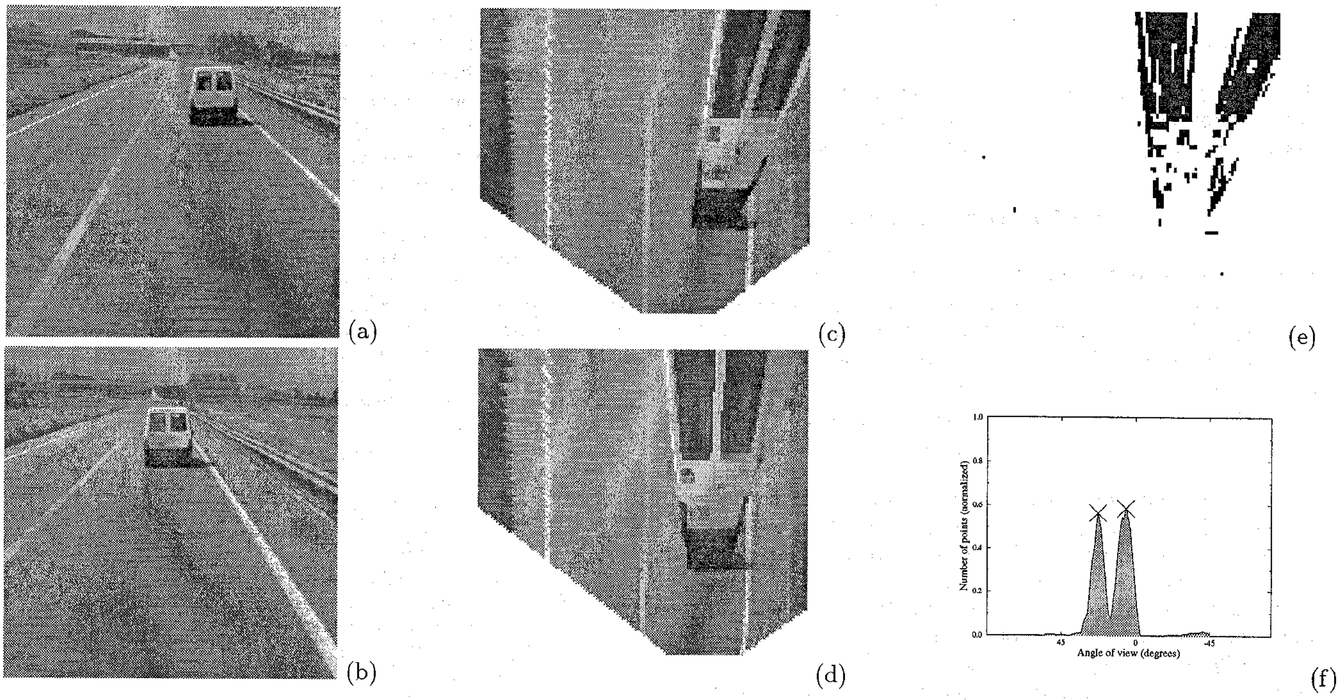

Figure 3: Obstacle detection in case of a single vehicle in front of the stereo system: (a) left image; (b) right image; (c) left reorganized image; (d) right reorganized image; (e) difference image; (f) polar histogram showing two maxima corresponding to a single object.

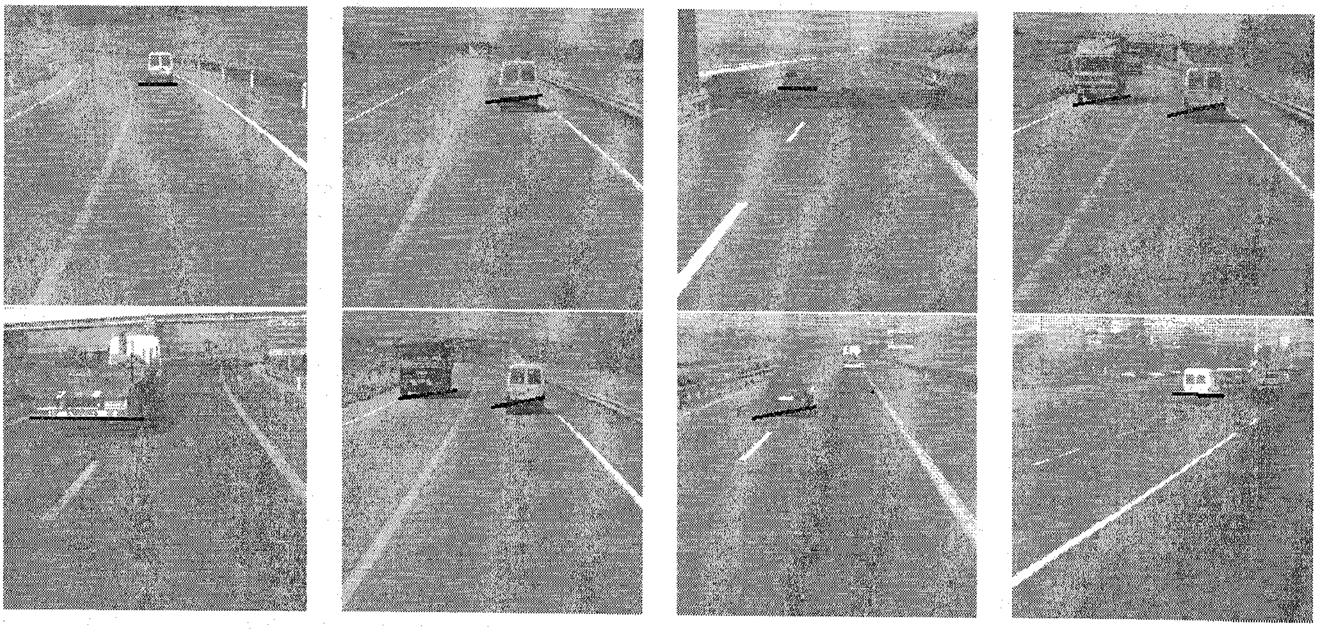

Figure 4: Results of the complete processing in different road conditions: a black marker indicates the obstacle's bottom