

The Evolution of a Massively Parallel Vision System for Real-Time Automotive Image Processing

Alberto Broggi*

Dipartimento di Ingegneria dell'Informazione
Università di Parma
Parma, ITALY, I-43100

Abstract

This paper presents the first prototype of the PAPRICA massively parallel system which is integrated on the MOB-LAB experimental land vehicle for real-time vision-based road markings detection. Its main bottlenecks are highlighted and its evolution toward a linear array is discussed. This system has been enhanced with a simple but powerful interprocessor communication network for the exchange of information among processors not directly connected, which allows an extremely efficient implementation of the road markings detection algorithm as well as other morphological applications.

1 Introduction and Motivation

This paper presents the design and evolution of a low-cost computer vision system for the real-time implementation of automotive applications. The algorithm considered in this work (*real-time road markings detection*) has been proved to be very effective thanks to its excellent robustness with respect to shadows and global illuminations changes on the road path, as well as to the vehicle's pitch during movement [3]. The whole system, capable of detecting road markings with a high confidence on extra-urban rural roads, has been extensively tested for more than 1500 km at speeds up to 60 kph on board of the MOB-LAB land vehicle, the MOBILE LABORATORY integrating the results of the Italian PROMETHEUS research Units.

Due to the perspective effect induced by the specific acquisition conditions, in the acquired image the road markings' width changes according to their distance from the camera. The removal of the perspective effect, if possible, causes each pixel to represent the same portion of the road; in this way the road markings' width is invariant with their position within the image.

In order to remove the perspective effect it is necessary to know the specific acquisition conditions (camera

position, orientation, optics,...) and the scene represented in the image (a structured¹ road, which is now assumed to be flat); this constitutes *a-priori knowledge*. The road markings detection algorithm can be conveniently divided into two steps: (i) the former (*expectation driven*, exploiting the a-priori knowledge) consists of an image transform aimed to remove the perspective effect; (ii) the latter (*data driven*, exploiting the sensorial data) is a mere low-level pattern matching.

Some considerations on both system performance and power consumption requirements [10, 11] show that a low-cost *special-purpose massively parallel* architecture is the solution that offers the possibility (i) to achieve a sufficiently fast (real-time) processing (which is of basic importance in the automotive field) and (ii) to integrate the hardware system on a moving vehicle (whose constraints on size and production- and operational-costs are extremely tight). Nevertheless, the SIMD system must be enhanced with an ad-hoc extension for the efficient implementation of the initial image transform.

The paper is organized as follows. The next Section briefly describes the image reorganization process aimed to remove the perspective effect. Section 3 describes the complete vision system, while Section 4 describes the hardware prototype (PAPRICA) currently installed on MOB-LAB. Section 5 highlights its architectural problems and critical points. Section 6 presents the evolution of the system toward the PAPRICA-3 linear array architecture. Section 7 ends the paper with some remarks.

2 The Inverse Perspective Mapping

The image reorganization is accomplished with a transform that loads into each pixel of the reorganized image the value of its corresponding pixel of the original one, according to a given projection function, acting as the dual of a *ray-tracing* algorithm [16].

Being easily and efficiently implementable on any mesh-connected SIMD massively parallel system, the subsequent identification of the road markings in

*This work was partially supported by CNR Progetto Finalizzato Trasporti under contracts 93.01813.PF74 and 93.04759.ST74. The author can be reached at the following e-mail address: broggi@CE.UniPR.IT

¹With painted road markings.

the *reorganized* image (based on morphological operations) does not suffer from the main problem of vision-based road detection systems: the presence of shadows on the road region [13, 14]. In the reorganized image, the road markings are represented by *quasi-vertical* lines of *constant width* that can be easily identified and extracted through a simple morphological algorithm (whose detailed description can be found in [5]) detecting horizontal black-white-black transitions. Figure 1 presents the intermediate results of the processing and the behavior of the algorithm in different road conditions.

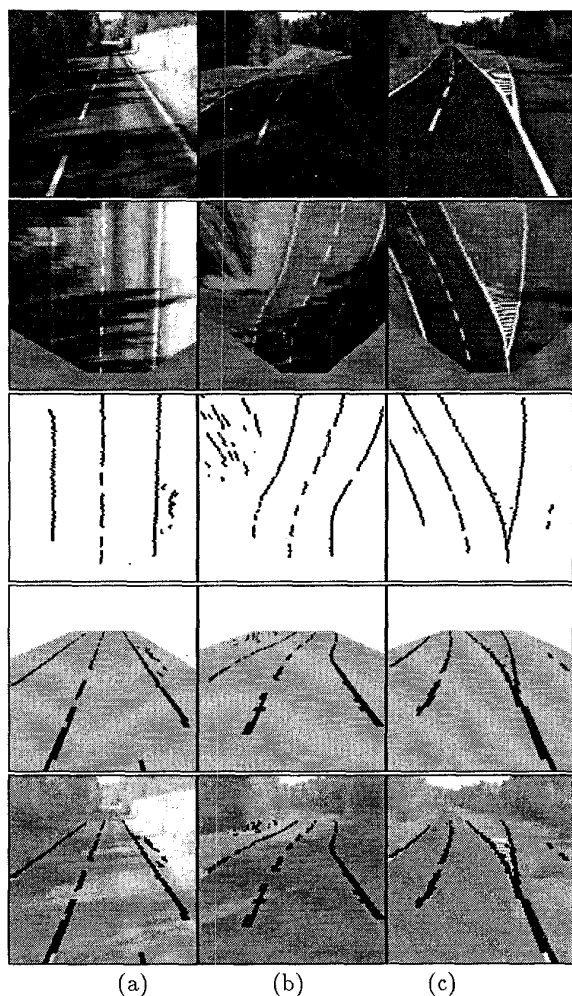


Figure 1: The detection of road markings in three different conditions: straight road in severe shadows conditions, curved road with shadows, junction. (a) input image; (b) reorganized image; (c) detection of black-white-black transitions in the horizontal direction; (d) reintroduction of the perspective effect; (e) superimposition of the result onto a brighter version of the original image.

3 The System Architecture

In order to fully understand the requirements of the hardware processor, it is of basic importance to focus our attention toward the architecture of the *complete* vision system. A camera acquires grey-tones images at a resolution of 512×512 pixels and pipelines them to the massively parallel processor which is in charge of the low-level portion of the processing. The result is then analyzed by a traditional serial system (hosted on the same common bus) which warns the driver in dangerous situations through a set of leds installed on MOB-LAB control-panel.

The detection of the road markings is just one of the functionalities that are currently performed by the system, and thus the reorganization process must take place only in the road markings detection module and not during the image acquisition phase or during the loading of the data into the low-level processor's memory. In fact, other processing modules (optical flow computation, vehicles detection, obstacle detection) [1] take as input sequences of *non-reorganized* images. The need to process *plain* images (and not only *reorganized* ones) does not allow to use an optical device (such as a properly shaped lenses system) to distort the incoming images, performing the reorganization process in the fastest possible way and with a higher resolution.

4 The current implementation

The system currently operative on MOB-LAB is configured as shown in Figure 2. PAPRICA (PARallel

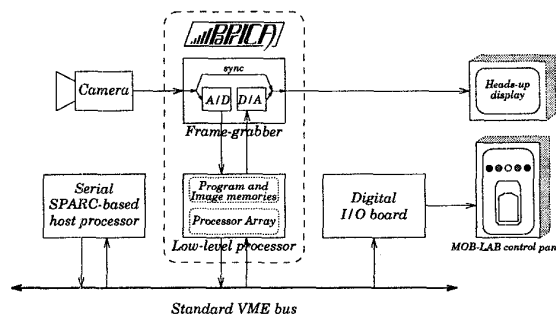


Figure 2: The configuration of the vision system

Processor for Image Checking and Analysis) [7] is a full-custom SIMD massively parallel system composed of a square matrix of 256 single-bit PEs disposed on the nodes of a 2D mesh, each one with full 8-neighbors connectivity [15, 12, 17]. In the current prototype the Processor Array (PA) is composed of an array of 4×4 full custom ICs ($1.5 \mu\text{m}$ CMOS, 45 mm^2 , ≈ 35000 transistors), each of them containing a sub-array of 4×4 PEs. Each PE has an internal memory composed of 64 bits; the image to be processed is thus stored in an external memory (Image Memory, IM) and an *external* PE virtualization mechanism is used [8]. Data

flow between the IM and the PA through a 16-bit data bus. The PAPRICA low-level processor, integrated on a single VME board (6U), comprises 5 major functional parts:

1. the Program Memory, up to 256k instructions;
2. the Image Memory, up to 8 MBytes of static RAM;
3. the Processor Array, with 256 single-bit PEs;
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); it can also be connected to an external monitor to show the intermediate and final results of the processing in real-time.
5. the Control Unit, managing the activities of the whole system.

The following subsection describes the architectural enhancement explicitly designed for an efficient image reorganization.

4.1 The data reorganization subsystem

The reorganization process consists of resampling the original image: some pixels may be moved to more than one new position (oversampling), while other pixels may be discarded (undersampling).

Since the source and destination coordinates for each pixel movement (communication) are fixed (once determined the correct camera calibration), they can be stored in a look-up table. In this way the reorganization process requires no run-time computation at all.

The solution adopted in the current prototype for a fast image reorganization is extremely simple and effective. A 2D look-up table (LUT) is stored by the host into PAPRICA memory; each one of its elements contains the IM address of its corresponding pixel in the input image. The reorganization process is then reduced to a set of indirect memory accesses. This solution did not require any additional component on the VME board: only the reprogramming of the FPGA (integrating the system controller) was needed. Moreover, the use of different LUTs allows to perform different reorganizations in the same program (such as the removal and reintroduction of the perspective effect, as shown in Figure 1).

The movement of a single pixel requires two memory readings (the address of its corresponding pixel in the input image from the LUT and the value itself) and a single memory store (the value in the reorganized image). Each memory access takes 50 ns (the IM is a 35 ns static RAM), thus the reorganization of a 128×128 image takes about $3 \times 50 \times 128^2 \simeq 2.5$ ms. Since the reorganization process is performed *on-board*, its execution is extremely fast; conversely it has the disadvantage that it cannot be overlapped with any other activity of the system (such as image acquisition or low-level processing).

5 Critical analysis of the current prototype

The evaluation of the system performance as well as some theoretical studies [8] have pointed out some architectural problems due to:

- the logical organization of the IM and data bus;
- the CISC implementation of the instruction set;
- the restricted neighborhood involved in morphological operations;
- the impossibility to overlap I/O and processing;
- the 'serial' implementation of the reorganization process.

These architectural issues, which have been theoretically addressed in [8] and validated in the analysis of many different applications [4, 5, 1], have led to (i) the choice of a linear array architecture with an ad-hoc interprocessor communication network; (ii) the redesign of the data-bus and memory word logical organization; and (iii) the decision to support a wider PE neighborhood.

6 Architectural Evolution: The **PAPRICA-3** System

The specifications of the new massively parallel SIMD system PAPRICA-3 [6] are now completely defined, while a system level simulator (running on traditional workstations) is available to test the architecture and the algorithm prior to developing the ICs.

As shown in Figure 3, the core of the processor is a dedicated SIMD cellular architecture based on a linear array of N identical single-bit PEs (in the first version $N = 256$). The array is connected to an external IM

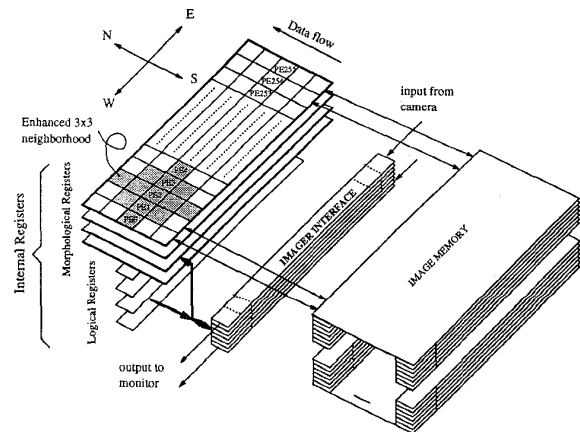


Figure 3: Structure of a 256 PEs version of PAPRICA-3

via a bidirectional N -bit data bus; therefore each memory access transfers a complete vector of N pixels at a

time, 1 bit per pixel. The rationale behind this system is that the size of the PA matches exactly the width of the input image. This solution allows to start the processing *before* the complete acquisition of the input image.

The PA is composed of two different kind of internal registers: Morphological Registers (MOR) and Logical Registers (LOR). The MOR consist of 5-bit south-to-north shift-registers whose southernmost position is loaded one image line at a time by the data bus. The presence of the shift-registers allows to perform morphological operations on an enhanced 3×3 neighborhood, which is shown in Figure 3. The LOR are used as a temporary storage and to perform logical operations.

The system features also a serial-to-parallel I/O device, called *IMager InterFace* (IMIN), connected to a conventional camera and to a monitor. While a line is processed, the IMIN behaves like a shift-register, serially loading the following image line from the camera. At the end of the processing, the PA loads the following line from the IMIN and eventually stores the results back again into the IMIN for the successive output on a monitor. During the data acquisition process, the IMIN serially loads the incoming data from the camera and simultaneously outputs the processed data serially to a monitor.

6.1 The data reorganization subsystem

An ad-hoc interprocessor communication mechanism, whose interconnection topology is fully reconfigurable at run-time, is available for exchanging information among PEs which are not directly connected. The Interprocessor Communication Network (ICN), shown in Figure 4, allows concurrent *line-wise* communications: each PE drives a switch that enables or disables

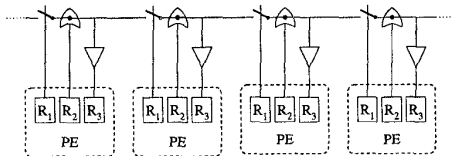


Figure 4: Interprocessor Communication Network (ICN): in each PE, Register R_1 drives its ICN switch; Register R_2 sends its value over the network; and Register R_3 collects the value from the network.

the communication (in wired-or) between itself and its left neighbor; adjacent PEs can thus be dynamically grouped into clusters. Each PE can broadcast its value to the whole cluster within a single instruction. This feature is extremely useful for the image reorganization process and for problems such as *seed-propagation* or labeling of connected components and in the handling of hierarchical (pyramidal) data structures [9, 18, 19, 2].

Unfortunately, the reorganization process mentioned above is based on *global*² communications: for

²Not necessarily within the *same* image line.

an efficient implementation on PAPRICA-3 the pixels movements must be within the same image line (*line-wise* reorganization), and thus an approximation is required for the independent reorganization of each line of the image.

6.2 Real-time line-wise image reorganization on

The ICN subsystem is used to group the PEs in clusters; each cluster performs a single communication between a source and as many destinations as required. For each set of *compatible communications*³, three different binary constants (images) are required:

- **Source:** specifying the PEs which act as sources in the current set of communications;
- **Path:** defining the grouping of PEs; since each PE drives a switch that connects itself to its left neighbor, the leftmost PE in a cluster must hold a zero value;
- **Destination:** identifying the destination PEs which must read and store in their local memory the value present on the ICN network.

Figure 5.a and 5.b present an oversampling and an undersampling mapping respectively, typical of the image reorganization process: in both cases the data movements can be exhaustively partitioned into 3 subsets of *compatible communications*.

Using the PAPRICA-3 system-level simulator it is possible to derive the time required for a cycle of communications, which is lower than $2.4 \mu s$. In the reorganization of a 128×128 pixel image, as shown in Figure 6, a maximum of 19 cycles per line is required. Every single image line is thus reorganized in less than $46 \mu s$. A commercially available off-the-shelf camera (512×512 pixels, 8 bit/pixel, 25 Hz) acquires a single line (512 pixels) in $80 \mu s$; thus, thanks to the possibility to overlap data I/O and processing, PAPRICA-3 allows to perform (i) the complete line reorganization and (ii) even a portion of the low-level processing *in real-time* during the acquisition of the subsequent image line.

7 Conclusions

This paper presented the evolution of the system which is in charge of the real-time low-level processing of images on board of the MOB-LAB land vehicle. The current PAPRICA prototype (a 2D mesh-connected SIMD architecture) includes a subsystem specifically designed to remove the perspective effect in the incoming image which can be triggered only *after* the complete image acquisition.

Some critical architectural problems have been highlighted and the consequent evolution of the system toward the PAPRICA-3 linear array architecture has

³Communications that can be performed *concurrently*.

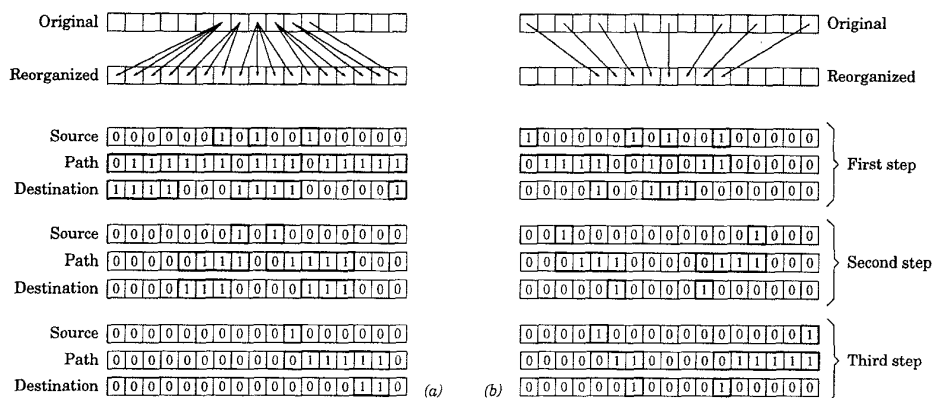


Figure 5: Line-wise (a) oversampling and (b) undersampling both performed in 3 steps

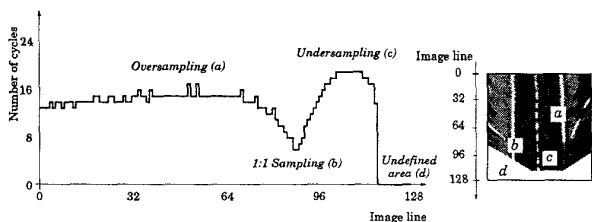


Figure 6: Number of communication cycles required to reorganize a 128 pixel line as a function of the position of the image line. In phase (a) (corresponding to a heavy oversampling) the number of iterations remains almost constant; it is reduced to a minimum when a quasi 1:1 mapping is performed (b); and it grows up again in the undersampling phase (c); finally it is reduced to zero in the lowest undefined region (d).

been presented. The new PAPRICA-3 system has been enhanced with a run-time reconfigurable PE interconnection network which allows multiple simultaneous communications among PEs not directly connected. This feature allows to perform the considered image reorganization process in real-time during data acquisition and can efficiently solve common problems, such as seed propagation, labeling of connected components, or the simulation of pyramidal interconnections between the PEs.

References

- [1] G. Adorni, A. Broggi, G. Conte, and V. D'Andrea. Real-Time Image Processing for Automotive Applications. In P. A. Laplante and A. D. Stoyenko, editors, *Real-Time Image Processing: Theory, Techniques and Applications*. IEEE Press and SPIE Press, Summer 1996. In press.
- [2] M. Albanesi, V. Cantoni, U. Cei, M. Ferretti, and M. Mosconi. Embedding Pyramids into Mesh Arrays. In *Reconfigurable Massively Parallel Computers*, pages 123–140, New Jersey, 1991. Englewood Cliffs.

- [3] A. Broggi. A Massively Parallel Approach to Real-Time Vision-Based Road Markings Detection. In *Procs IEEE Intelligent Vehicles'95*, pages 84–89, Detroit, 1995.
- [4] A. Broggi. Parallel and Local Feature Extraction: a Real-Time Approach to Road Boundary Detection. *IEEE Trans on Image Processing*, 4(2):217–223, February 1995.
- [5] A. Broggi and S. Bertè. Vision-Based Road Detection in Automotive Systems: a Real-Time Expectation-Driven Approach. *Journal of Artificial Intelligence Research*, 3:325–348, December 1995.
- [6] A. Broggi, G. Conte, G. Burzio, L. Lavagno, F. Gregoretti, C. Sansoè, and L. M. Reyneri. PAPRICA-3: A Real-Time Morphological Image Processor. In *Procs IEEE Intl Conf on Image Processing*, pages 654–658, Austin, TX, 1994.
- [7] A. Broggi, G. Conte, F. Gregoretti, C. Sansoè, and L. M. Reyneri. The PAPRICA Massively Parallel Processor. In *Procs IEEE Intl Conf on Massively Parallel Computing Systems*, pages 16–30, Ischia, Italy, 1994.
- [8] A. Broggi and F. Gregoretti. Performance Evaluation and Optimization in Low-Cost Cellular SIMD Systems. In *Journal of Systems Architecture*, 1995. In press.
- [9] V. Cantoni and M. Ferretti. *Pyramidal Architectures for Computer Vision*. Plenum Press, London, 1993.
- [10] A. Chandrakasan, S. Sheng, and R. Brodersen. Design considerations for a future multimedia terminal. *Proc. WIN-LAB Workshop - Rutgers Univ.*, New Brunswick, NJ, 1990.
- [11] G. H. Forman and J. Zahorjan. The Challenge of Mobile Computing. *Computer*, 27(4):38–47, April 1994.
- [12] T. Fountain. *Processor Arrays: Architectures and applications*. Academic-Press, London, 1987.
- [13] T. M. Jochem and S. Baluja. Massively Parallel, Adaptive, Color Image Processing for Autonomous Road Following. In H. Kitano, editor, *Massively Parallel Artificial Intelligence*. AIII Publishers in cooperation with MIT Press, 1993.
- [14] S. K. Kenue. Correction of Shadow Artifacts for Vision-based Vehicle Guidance. In *Procs of SPIE - Mobile Robots VIII*, volume 2058, pages 12–26, 1994.
- [15] NCR Corporation, Dayton, Ohio. *Geometric Arithmetic Parallel Processor*, 1984.
- [16] W. M. Newman and R. F. Sproull. *Principles of Interactive Computer Graphics*. McGraw-Hill, Tokyo, 1981.
- [17] S. Reddaway. DAP - A Distributed Array Processor. In *1st Annual Symposium on Computer Architectures*, pages 61–65, Florida, 1973.
- [18] A. Rosenfeld. *Multiresolution Image Processing and Analysis*. Springer Verlag, Berlin, 1984.
- [19] S. L. Tanimoto and K. Kilger. *Structured Computer Vision: Machine Perception through Hierarchical Computation Structures*. Academic Press, NY, 1980.