REAL-TIME OBSTACLE DETECTION ON A MASSIVELY PARALLEL LINEAR ARCHITECTURE

Massimo Bertozzi, Alberto Broggi, Alessandra Fascioli
Dipartimento di Ingegneria dell'Informazione
Università di Parma, I-43100 Parma, Italy
{bertozzi,broggi,fascal}@CE.UniPR.IT

This paper presents a real-time solution to the problem of obstacle detection in automotive applications using image processing techniques. To speed-up the processing a massively parallel engine has been used and the algorithms tuned to match the specific features of the computing architecture. The system acquires pairs of stereo images, checks for correspondences, and remaps the resulting image in a new domain to ease the following processing steps. The whole processing is performed on PAPRICA-3, a massively parallel system whose processing elements are disposed on a linear array; the proposed system allows to reach video rate performance. The whole system is currently simulated and is expected to be available by mid '98.

1 Introduction

Vision plays a basic role in vehicular applications, in which it can be used to sense the environment without interfering with other vehicles' sensors: in our system [3] two cameras installed on board of a moving vehicle acquire sequences of stereo images which are fed to a computer system for processing. The functionality addressed in this work is the localization of potential obstacles on the path of the moving vehicle.

In the GOLD (Generic Obstacle and Lane Detection [2]) system, obstacle detection is reduced to the determination of the *free space* in front of the vehicle, namely the portion of the road where the vehicle can safely move. The algorithm, depicted in figure 1.a, is briefly summarized in the following:

- a pair of grey-level images is acquired by a stereo vision system;
- thanks to the assumption of a flat road in front of the vehicle [9] and to the knowledge of the vision system parameters, the perspective effect can be removed from both images: a geometrical transform is used to convert the incoming images in a new domain (road domain), producing two *remapped images*; this step, aimed to recover the road texture, is of paramount importance since it eases the following processing steps [2];
- in case no obstacles are present in the scene the two remapped images represent a bird's eye view of the same patch of the road; on the contrary, the presence of obstacles invalidates the flat road hypothesis, thus producing disparities in the two remapped images;

(0-7803-4229-1/97/$10.00 1997 IEEE)

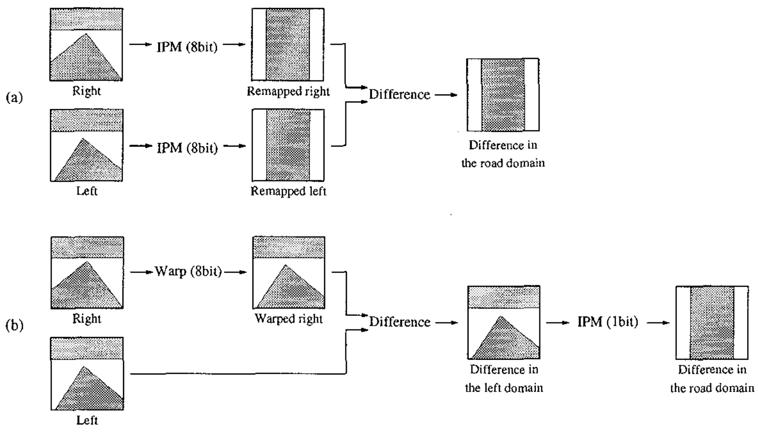
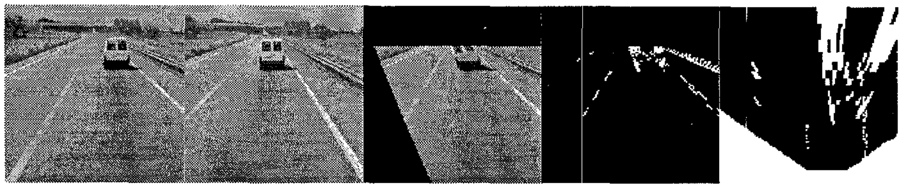


Figure 1: the difference between (*a*) the original algorithm and (*b*) its PAPRICA-3 version

- in order to determine the presence of obstacles and to estimate both their distance and angle of view, the difference between the two remapped images is computed (*difference image*), binarized, and fed to a high level processing module.

The PAPRICA-3 [6] architecture, currently under fabrication, has been developed to address this problem. It is composed of a linear array of 128 PEs devoted to the simultaneous processing of a single image line and features a direct I/O camera interface for real-time image acquisition. To take advantage of its architecture, the obstacle detection algorithm was redesigned to match the features of the new system. Due to the characteristics of a line-wise processing implemented on PAPRICA-3, a more effective algorithm has been developed.

As in the approach discussed above, the system starts acquiring a pair of grey-level images; since the geometrical transform used to remove the perspective effect requires the execution of complex data transfers between non-adjacent pixels, its implementation on PAPRICA-3 would suffer from a low efficiency. Therefore only one image (the right one) is processed using a different geometrical transform: instead of removing the perspective effect, the right image is *warped* onto the left one. Assuming all the acquisition parameters known and a flat road in front of the vehicle, this warping produces a new image (*warped image*) that represents the scene as framed from the position of the other camera. In case the flat road hypothesis is not met, the left and the warped images differ. Since the following processing steps are simplified when the input image is remapped in the road domain, the perspective effect



(a) (b) (c) (d) (e)

Figure 2: (*a*) and (*b*) show the left and right acquired images respectively, (*c*) presents the right image warped onto the left one, (*d*) shows the thresholded difference image, while (*e*) depicts the difference image remapped onto the road plane.

is then removed from a binarized version of the difference image (1 bit deep). The analysis of this result leads to the determination of the free space in front of the vehicle.

The advantages of this approach (summarized in figure 1.b) are the following:

- the warping of the right image can be approximately performed by a set of line shifts which match a basic feature of PAPRICA-3;
- the removal of the perspective effect is performed only on one binary image (the difference image) instead of on two 8 bit deep grey level images (left and right images).

As an example, figure 2 shows the sequence of images generated by this algorithm.

2 The Computer Architecture

The core of the PAPRICA-3 [6] (PArallel PRocessor for Image Checking and Analysis, version 3) system is a dedicated SIMD cellular architecture based on a linear array [8] of Q identical 1-bit Processing Elements (PEs). The array as a whole is connected to an external image memory via a bidirectional Q-bit data bus; therefore each memory read or write operation transfers a complete vector of Q pixels at a time, 1 bit per pixel. This specific organization of the data-bus (Q bits wide and 1 bit deep) allows to reach a hardware efficiency higher than in the first prototype architecture, since the whole set of data transferred within a single cycle is generally completely significant.

The rationale behind this system is that the size of the PA matches exactly the width of the input image. This solution reduces the PE virtualization mechanism problem, which has been proven to be a critical design issue. The PA processes one full image line per machine cycle, whose duration ranges from

10 ns to 40 ns, depending on the specific instruction flow. Data are transferred into the PEs internal registers, processed, and explicitly stored back again into the external memory according to a RISC-oriented processing paradigm.

The system comprehends also a serial-to-parallel I/O device, called *Imager Interface*, connected to a conventional camera. While a line is processed by the PA, the Imager Interface automatically loads the following image line from the camera. At the end of the processing, the PA stores in parallel the results back into the Imager Interface (on different bit-planes) and loads in parallel the following image line. During the data acquisition process, the Imager Interface behaves like a shift-register, loading the data serially from the camera, and outputs the processed data serially to a monitor.

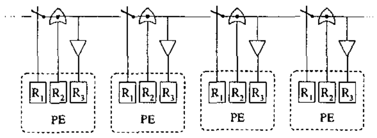


Figure 3: Interprocessor Communication Network (ICN): for each PE, Register R_1 drives its ICN switch; Register R_2 sends its value over the network; and Register R_3 collects the resulting value from the network.

An interprocessor communication mechanism has been included to exchange information among PEs which are not directly connected: the Interprocessor Communication Network (ICN), shown in figure 3. It allows global and multiple communications among components of different subsets of the PA (clusters of adjacent PEs) and its interconnection topology is fully and dynamically programmable: each PE drives a switch that enables or disables the communication (in wired-or) between itself and its left neighbor; the PEs can thus be dynamically grouped into clusters in which each PE can broadcast its value to the whole cluster within a single instruction. This feature is of basic importance for the application discussed in this work.

3 Image Warping

This section describes the warping procedure, namely the geometrical transform that, starting from the right image produces a new image that represents the road region in front of the vehicle as it were observed from the left camera.

The application of this transform requires the knowledge of the following parameters (see fig 4): *(i)* the height h and distance d of the two cameras; *(ii)* the viewing directions: for each camera the optical axis $\hat{o}$ is determined by the following angles: $\overline{\gamma}$, the angle formed by the projection (defined by versor $\hat{\eta}$)

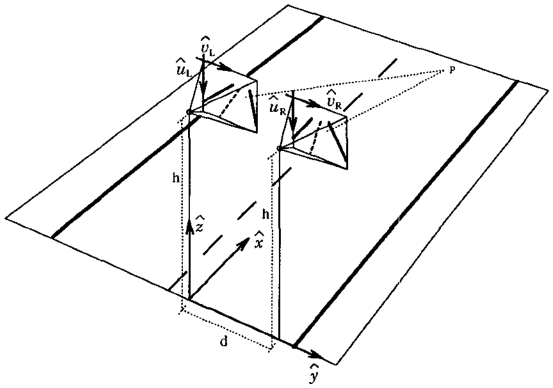


Figure 4: The geometry of the stereo vision system

of the optical axis $\hat{o}$ on the plane $z = 0$ and the x axis; and $\bar{\theta}$, the angle formed by the optical axis $\hat{o}$ and versor $\hat{\eta}$; *(ii)* the cameras angular aperture 2α; *(iv)* the cameras resolution $n \times n$.

Considering $\bar{\theta}_L = \bar{\theta}_R = \bar{\theta}$, the relationship between the coordinates (u_R, v_R) in the right image and the coordinates (u_L, v_L) in the left image is given by:

$$\begin{cases} u_R \approx u_L \\ v_R \approx v_L + \frac{n-1}{2\alpha} \left\{ \bar{\gamma}_L - \bar{\gamma}_R - \frac{d}{h} \mathrm{tg} \left[(\bar{\theta} - \alpha) + u_L \frac{2\alpha}{n-1} \right] \right\} = v_L + f(u_L) \end{cases} . \tag{1}$$

The procedure defined by equation (1) produces a left view of the road plane starting from the right view. It can be noticed from the analysis of equation (1) that the warped image can be obtained by simply translating each line of the right image and that the amount of the translation depends on the row number.

Assuming a flat road and no obstacles in front of the vehicle, the left image and the warped one represent the *same* texture. Any difference between these two images (see fig. 2.d) represents a deviation from the assumption of flat road, and thus identifies an obstacle, i.e. anything rising up from the road surface. The following high level obstacle detection process is thus based on the analysis of the difference image.

3.1 Parallel Implementation on PAPRICA-3

Since the warping procedure consists of a east line shift operation, it can be efficiently performed by PAPRICA-3 which is based on a linear array of PEs whose length matches the image width. Each PE obtains data thanks to a direct connection to its WEST and WEST WEST neighbors.

Thanks to the knowledge of the acquisition system setup, obtained by means of a calibration phase, the shift values for each image line are known at compile time and have been hardcoded in the PAPRICA-3 program, thus leading to an execution time of about 0.2 ms for a 128×128 pixel 8 bit image.

4 Inverse Perspective Mapping

After the determination of the disparities between the warped and the left image, the resulting image is binarized and remapped in the road domain. The remapping process is accomplished by applying the *Inverse Perspective Mapping* (IPM) [9] transform. Its main advantage is that in the remapped image the information content is homogeneously distributed among all pixels; this fact allows to perform efficiently the following processing steps on a SIMD system, such as noise cleaning morphological filters.

The implementation of this transform requires the movement of pixels to/from arbitrarily positions within the same image line; moreover, the value of some pixels may be duplicated and sent to more than one new position, while the values of other pixels may be discarded. Fortunately, these movements, directly derived from the setup of the acquisition system, are completely known at compile-time and thus can be hardcoded into the program.

4.1 Line-wise Global Communications

To perform these pixels movements in parallel, a specific reconfigurable interconnection network (ICN, see section 2) has been included in PAPRICA-3. This kind of communications can be efficiently performed by the ICN network [4], but in general the ICN cannot perform all global communications simultaneously. Each PE can be the source for one or more communications, but obviously it can be the destination of a single communication only. Thus this process must be serialized in independent steps, each step involving only *compatible* communications, namely that can be performed concurrently. An algorithm [5] is thus required in order to determine the optimal partitioning of the set of communications into the smallest number of subsets.

4.2 Parallel Implementation on PAPRICA-3

The execution of a single data transfer between a source and as many destinations as required needs the exclusive allocation of a cluster of adjacent

PEs. Thus, for each set of compatible communications, three different binary constants (image lines) are defined:

- **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 collected from the ICN network.

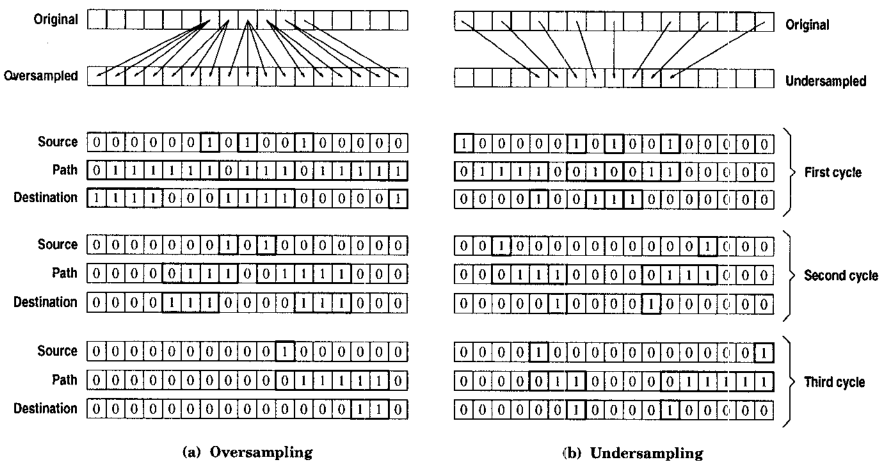


Figure 5: Two cases of line-wise global communications used by IPM: oversampling and undersampling

Figures 5.a and 5.b present oversampling and undersampling mappings; in both cases the data movements can be exhaustively partitioned into 3 subsets of compatible communications (requiring 3 cycles of communications).

A single cycle of simultaneous communications requires the execution of about 30 Assembly instructions. Since PAPRICA-3 internal structure[7] is based on a fairly complex pipeline, a specific ordering of the assembly instructions (*instruction scheduling* [10]) has been considered, using both traditional techniques [1] and innovative approaches, such as a stochastic code optimizer. In the worst case (depending on the specific set of compatible communications) the PAPRICA-3 system takes about 200 ns for the execution of a single set of binary communications.

In the remapping of a 128×128 pixel image, a maximum of 19 cycles per line are required. Every single line is thus remapped in less than $\simeq 4\ \mu s$, thus the total time required to remap a 128×128 pixel binary image is about 0.5 ms.

5 Conclusions

In this work the parallelization of the obstacle detection functionality has been addressed; it is based on image warping followed by a geometrical transform. Since in the automotive field the processing speed is of basic importance, a special-purpose massively parallel architecture, PAPRICA-3, has been used as the computing engine to reach real-time performance. The execution of the portion of low level obstacle detection presented in this work (image warping and geometrical transform) takes less than 1 ms, thus allowing to process images at video rate.

Acknowledgments

This work was partially supported by the Italian National Research Council (CNR) under the frame of the Progetto Finalizzato Trasporti 2.

References

1. A. Aho, R. Sethi, and J. Ullman. *Compilers: Principles, Techniques, and Tools.* Addison-Wesley, Reading, MA, 1986.
2. M. Bertozzi and A. Broggi. GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. *IEEE Trans on Image Processing*, in press.
3. M. Bertozzi and A. Broggi. Vision-based Vehicle Guidance. *IEEE Computer*, 30(7):49–55, July 1997.
4. A. Broggi. The Evolution of a Massively Parallel Vision System for Real-Time Automotive Image Processing. In *Proceedings IEEE IPPS'96 - International Parallel Processing Symposium*, pages 724–728, Honolulu, HI, 1996. IEEE Computer Society.
5. A. Broggi. Global Communications on a Linear Array Architecture. *Journal of Parallel Algorithms and Applications*, 11(1-2):27–43, 1997. In press.
6. 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(2):114–136, 1997.
7. F. Gregoretti, F. Intini, L. Lavagno, R. Passerone, and L. M. Reyneri. Design and Implementation of the Control Structure of the PAPRICA-3 Processor. In *Proceedings 4^{th} EuroMicro Workshop on Parallel and Distributed Processing*, Braga, Portugal, January, 24-26 1996. IEEE Computer Society - EuroMicro. In press.
8. D. Helman and J. Jájá. Efficient Image Processing Algorithms on the Scan Line Array Processor. *IEEE Trans on PAMI*, 17(1):47–56, 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. B. Ramakrishna Rau and J. A. Fisher. Instruction Level Parallel Processing: History, Overview, and Perspective. *The Journal of Supercomputing*, pages 9–50, 1993.