Reconfigurable architectures for machine perception. An approach for autonomous vehicle navigation

Jose A. Boluda, Fernando Pardo, Joan Pelechano

Departament d'Informática. Universitat de València.
C/ Doctor Moliner, 50. 46100 Burjassot, SPAIN.
Email: {Jose.A.Boluda, Fernando.Pardo, Joan.Pelechano}@uv.es

Abstract

This paper deals with one of the current research fields developed by the computer architecture group at the University of Valencia. The work presented is concerned about the design of a reconfigurable module, designed with programmable logic, which can implement several interesting robot navigation algorithms. The use of reconfigurable logic in autonomous vehicles can be a good approach for achieving hardware speed for some intensive computations. It will be also possible, with this approach, to have software reconfigurability; that is, the same hardware can implement several different algorithms. In this way the autonomous vehicle will choose the most adequate algorithm for the task that it is going to carry out.

1 Introduction

The problem of the autonomous vehicle navigation in non-structured environments is a very complex task. The robot should detect risk of colliding with obstructions or with other moving vehicles. Moreover the platform may be stopped and it should detect if there is any approaching object that can collide with it.

A typical way for doing these tasks is to incorporate a visual system into the vehicle which implements several vision algorithms, like optical flow computation, time-to-impact computation, or differential algorithms for motion detection. These tasks are very computer intensive [Bea95] and should be performed with real-time restrictions, meaning that many times a hardware solution is required. Another problem is the amount of data that should be processed, a lot of memory is required in order to process images with a minimal resolution and quality.

In the other hand, a custom hardware module is not a flexible solution. The robot may need several different high-speed algorithms optimized for different situations: detecting motion from the stopped platform, detecting static obstacles from the platform with ego-motion or detecting movement independent of the platform movement. To provide the vehicle with custom hardware modules for each task is not a possible solution due to the cost, weight, size and power consumption.

A reconfigurable module for autonomous vehicle-navigation has been designed and is being constructed at the University of Valencia. The module is a pipe of Processing Elements (PEs) which are externally identical (pin-out and communications protocol between them). The overall algorithm is split into several PEs, being the number of the PEs, which form the pipe, scalable. SRAM based FPGAs have been utilized in order to achieve the reconfigurability of each PE.

The algorithms that the module can perform can be defined as fine-grain differential algorithms. This point will be clearer with the module architecture definition.

In the other hand a non-common visual system has been utilized. The log-polar formalism has shown to be especially interesting for robotic applications [Tis93]. The recent availability of a CMOS log-polar sensor and a log-polar camera developed from this sensor makes possible the use of these advantages in real applications [Par97] [Par98].

The module designed combine these two new technological approaches: A reconfigurable and modular architecture which can implement a wide range of fine-grain differential algorithms, and the log polar vision system which reduce the total amount of data to be processed and simplifies several image processing algorithms.

2 Precedent approaches

An example of the use of reconfigurable logic for image processing is the *Spectrum Reconfigurable Computing (RC) Platform* from Giga Operations Corporation [Tay96]. This platform can have up to 16 data processing modules, with 32 FPGAs, 128 MB of DRAM and 4MB of SRAM each, a really big quantity of hardware resources conditioned by the size of the Cartesian images. Several preprocessing algorithms (not very complex) as 3*3 convolution or nonlinear filters have been implemented working up to 60 MHz.

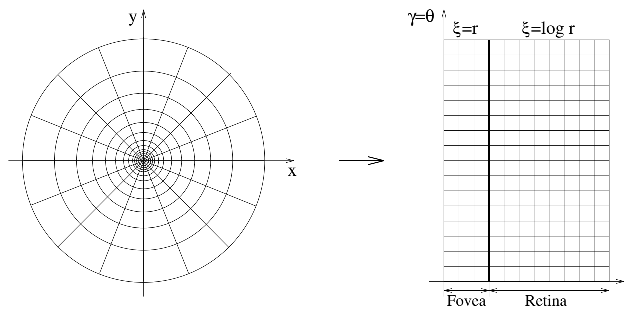
Another approach to a reconfigurable machine has been made with the well-known Splash-2 machine. This machine is as big as the Spectrum machine, with a general-purpose architecture, which can implement a wide range of algorithms [Bue96]. In this way, several image-processing algorithms have been implemented. The most interesting one up to date for robotic applications is the hierarchical visual tracking tested in a Sun workstation as host system [Pud97]. The big quantity of hardware resources needed to implement this algorithm in real time is due again to the big quantity of images to be processed.

Our approach, mixing a pipelined reconfigurable architecture with the log-polar vision formalism, will simplify a wide range of visual algorithms useful for robotic navigation. The processing module will be cheap and light enough for its inclusion as visual processing module in an autonomous navigation system.

3 Log-Polar vision and algorithms

In humans, the retina exhibits a non-uniform photoreceptor distribution: more resolution at the center of the image and less at the periphery, which allows a selective reduction of information. The figure 1 shows the layout of the foveal sensor that is included in the robotic camera.

Using this formalism it is possible to simplify and develop several vision algorithms which can be useful for robotic navigation. Each image has 9.5 Kbytes allowing enough resolution at the center of the image and keeping a wide field of vision. The equivalent resolution of the fovea is more than a Cartesian camera of 512x512 pixels that would require 256 Kbytes.



$$\begin{cases} \xi = \begin{cases} r & r \le r_0 \\ \log r & r > r_0 \end{cases} \quad r = \sqrt{x^2 + y^2} \\ \theta = \theta \end{cases}$$

Fig. 1. The log-polar transformation

3.2 Motion Detection

For an autonomous platform navigating in a non-structured environment it is very interesting to detect other moving objects. If the robot is stopped the external motion can be easily detected subtracting images. This is a simple approach that is not easier using the log-polar camera. The only advantage is that there is less data to compute and store.

The problem of detecting moving objects from a moving platform is much more complex. It is necessary to isolate the image displacement due to the platform egomotion from the image changes due to moving objects. Motion estimation techniques fit into three categories: feature-based, optical flow, and differential techniques. In any case, this is a very computational intensive problem.

The use of the log-polar coordinates makes possible the use of a differential technique when the movement is a lineal translation in the direction of the optical axis [Bol97]. This restriction can be assumed without any problem if the robot has a pan-tilt head. After the acquisition of the images E(ξ, θ, t), the algorithm can be split in the following pipelined stages:

1. Low-pass filtering for image smoothing.
2. First order temporal differentiation.
3. Second order temporal differentiation and thresholding.

The simplicity of the computations and the low amount of data to be processed make possible a hardware implementation with a pipelined architecture that will be described later.

3.3 Time to impact computation

Another interesting algorithm for autonomous navigation is the time-to-impact computation. In a non-structured environment to avoid collisions with static obstacles is also a very time-consuming task, that usually should be done with real time constrains. The use of the log-polar coordinates simplifies the computations and reduces the amount of data to be processed [Tis93].

In this case this algorithm can be divided into the following pipelined stages:

1. Low-pass filtering for image smoothing

2. Gradient and First order differential computation.
3. Integer division.

4 Module Architecture

The most interesting thing of our approach is the combination of the log-polar vision system (which simplifies the hardware needed) and the possibility of reconfiguration for the module. In this sense a SRAM based programmable logic will be needed. The platform will reconfigure the programmable devices which form the pipelined architecture downloading a stream of bytes for each pipe stage. This will fix the technology of the devices.

Also a certain complexity (number of registers and equivalent gates) will be required. The algorithms are pipelined and split into different stages, this characteristic reduces the number of registers for each processing element, but a medium-high complexity is required; that means FPGAs. Finally, the real-time restrictions of the algorithms force to choose a non-pure FPGA architecture with carry look-ahead and good velocity grades. The family of SRAM-based FPGAs FLEX8000 from ALTERA has been chosen to design the different processing elements.

A byte-level pipelined architecture has been designed for implementing fine-grain differential algorithms. The two algorithms described before have been divided into 3 pipelined stages and each stage has been designed using VHDL and fitted into a single SRAM FPGA.

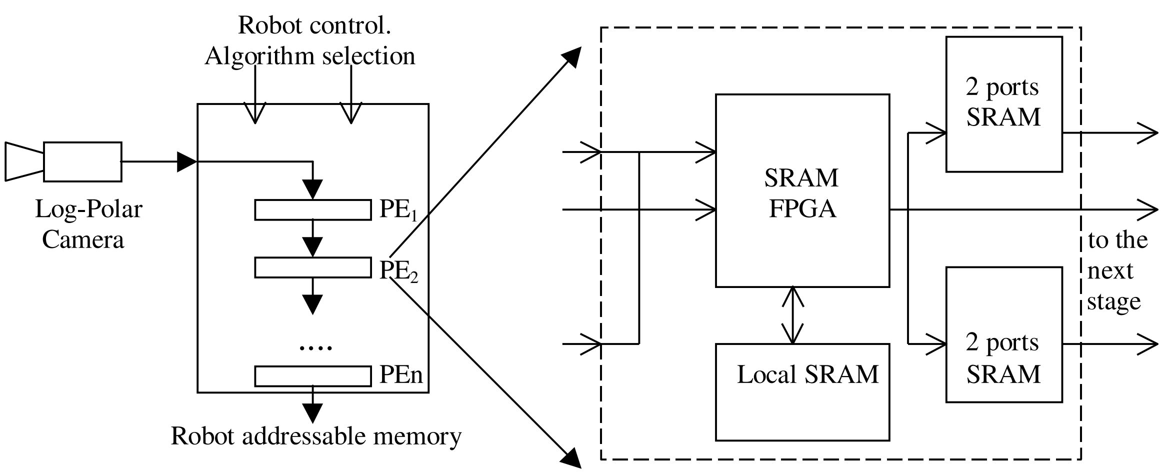


Fig. 2. Module and PE structure

Each FPGA has exactly the same pin-set and follows the same data-request data-ready protocol at byte level. In this way, all the FPGAs are externally identical. The different functionality is achieved when the master of the system (the robot control) decides which algorithm is going to use. Thus, it programs all the stages downloading its programming bits.

Each processing stage has a 16 Kbytes SRAM memory for local computations and 2 double-port 16 Kbytes SRAM memories as frame grabber for the differential algorithms. The small size of the log polar images makes possible the use of this memories. The use of double-port memories as intermediate frame grabbers will increase the pipe performance.

Each PE has a small PCB and a data-in and data-out connector for implementing the algorithm pipe. The PE pin-set, the communications protocol definition and the number of PEs needed are general enough for covering a wide range of interesting algorithms for robotics. All this small PEs are attached to the motherboard module (log-polar frame grabber) which controls the camera and gives the processed images to the robot (time-to-impact map, binary images with moving objects, etc.). The motherboard also receives the messages for reconfiguring the module and changes the FPGAs programming. Figure 2 shows the module, the processing element structure and the data flow in the pipe. The image data flow from left to right and the data-request data-ready protocol controls the image flow from right to left.

The two algorithms here described and other differential algorithms have been fully designed and simulated after its VHDL compilation. The PCB prototypes for the motherboard and for the PEs are now being constructed, and will be tested in an autonomous navigation platform as soon as they are ready.

5 Acknowledgements

The Generalitat Valenciana under project GV97-TI-05-27 is supporting this work.

6 References

[Bea95] *S. S. Beauchemin and J. L. Barron*. The computation of the optical flow. ACM computing surveys, Vol. 27, Nº 3. September, 1995.

[Bol97] *J. A. Boluda, J. Domingo, F. Pardo, and J. Pelechano*. Detecting motion independent of the camera movement through a log-polar differential approach. Computer Analysis of Images and Patterns. Lectures Notes in Computer Sciences Series, Vol. 1296, pp. 702-709. Kiel, Germany. September 1997.

[Bue96] *D. A. Buell, J. M. Arnold, W. J. Kleinfelder*. Splash 2: FPGAs for Custom Computing Machines. IEEE Computer Society Press. Los Alamitos. 1996.

[Par97] *F. Pardo, B. Dierickx and D. Scheffer*. CMOS Foveated image sensor: Signal scaling and small geometry effects. IEEE Transactions on Electron Devices. pp. 1731-1737, Vol. 44, No. 10. October 1997.

[Par98] *F. Pardo, B. Dierickx and D. Scheffer*. Space-Variant non-orthogonal structure CMOS image sensor design. IEEE Journal of Solid-State Circuits. Vol. 33, No. 6, June 1998.

[Pud97] *B. Pudipeddi, A. L. Abbott and P. Athanas*. Real-time hierarchical visual tracking using a configurable computing machine. Proceedings of the Fourth IEEE International Workshop on Computer Architecture for Machine Perception, CAMP'97. pp. 153-157. Cambridge, Massachusetts. October 1997.

[Tay96] *B. Taylor*. DSP filters in FPGAs for image processing applications. High-Speed Computing, Digital Signal Processing and Filtering Using Reconfigurable Logic. SPIE Proceedings, pp. 100-109, Vol. 2914. Boston, Massachusetts, USA. November 1996.

[Tis93] *M. Tistarelli and G. Sandini*. On the advantages of polar and log-Polar mapping for direct estimation of time-to-impact from optical flow. IEEE Transactions on Pattern Analysis and Machine Intelligence. pp. 401-409, Vol. 15, No. 4. April 1993.