

A C80 DSP-based Active Vision System for Real-Time Tracking¹

C. Guerra-Artal, M. Castrillón-Santana, J. Hernández-Sosa, J. Isern-González, J. Cabrera-Gómez, F.M. Hernández-Tejera

Edif. Dptal. Informática y Matemáticas - Campus de Tafira - 35017

Las Palmas de Gran Canaria - Gran Canaria - SPAIN

cayetano@mozart.dis.ulpgc.es

1. Introduction

Active Vision Systems can be considered as dynamical systems which close the loop around artificial visual perception, controlling camera parameters, motion and also controlling processing to simplify, accelerate and do more robust visual perception. Research and Development in Active Vision Systems [Aloi87], [Bajc88] is a main area of interest in Computer Vision, mainly by its potential application in different scenarios where real-time performance is needed such as robot navigation, surveillance, visual inspection, among many others. Several systems have been developed during last years using robotic-heads for this purpose. Most of them have based their design on specific hardware, using transputers or DSPs networks, commonly, using VME bus to interconnect the system. [Pahl93][Seel96].

In this paper, an Active Vision System is presented. This system has been developed made up by off-the-shelf components: DSPs, Pentium processor and a stereoscopic robotic-head. These elements have allowed us to design and build a cost-effective system whose first prototype is able to perform detection and tracking of mobile objects in real-time and continuous operation.

2. Hardware

2.1 Initial Considerations

In trying to accomplish with real-time performance requirements, several technologies are available:

Custom VLSI design: This technology is expensive, not reusable and it is only justified when other main-stream technologies lag behind the required performance in an order of magnitude or more or volume production justifies its development costs.

Configurable Devices such as FPGAs: It is a very attractive option as the same piece of hardware can be reconfigured to target a different functionality. However, it is still at an early stage of development and its usage requires experience with VHDL and other design and development tools.

Digital Signal Processors or transputers: DSPs and transputers have been the common off-the-shelf resources when trying to increase the computational power available 'in a box'. Numerous active vision systems have been developed using transputers and some others utilize DSPs offering high speed multiport communication like the TMS320C40.

General purpose processors: Taken into account its availability, low cost and sustained increase in performance, this type of processors are preferred when the computational requirements of the problem at hand does not justify the utilization of other more expensive alternatives.

¹This work has been partially supported by Spanish CICYT Project TAP9-0288

Several computer vision systems have been developed using all these technologies. Custom VLSI design, although still an alternative, is being replaced by FPGA-based designs which offers also high performance in a much more flexible context. However, DSP and transputer based designs have been the election for development of experimental active head-eye systems as they offer a scalable computational power at a fraction of the cost of competing technologies. The launch into the market of TMS320C80 can be considered as a major breakthrough in this line of development basically by its novel architecture and performance.

2.2 System Hardware

The main feature of the system presented is the design of a conceptual architecture based on the interaction of several hardware components to compose a full-fledged perception-action system. Each subsystem makes use of a particular hardware.

The perception subsystem consists on a couple, one for each *eye*, of TMS320C80 development board to perform image acquisition and processing.

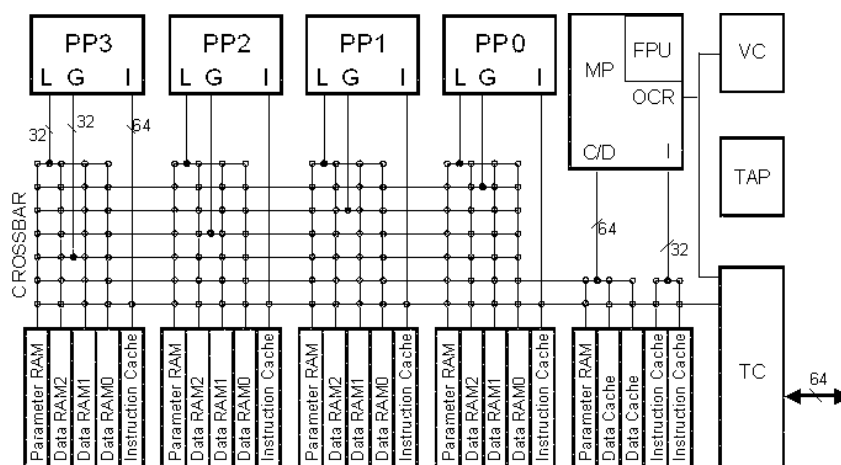
The action subsystem is a commercial motorized robotic head that offers four degrees of freedom: pan, tilt and two vergences, plus other six optical degrees: iris, zoom and focus for both lenses.

It is obvious that both perception and action hardware need to be interconnected in order to build a closed-loop system. Results coming from the perception subsystem must be translated into commands for the action subsystem, i. e., a movement in the followed target will be reflected in the robotic head pose. A PC running Windows NT 4.0 provides a suitable interconnection layer for all this hardware.

2.3 TMS320C80 Architecture

The TMS320C80 integrated circuit is a digital signal processor (DSP) designed by Texas Instruments, which offers a great performance. The TMS320C80 integrates onto a single chip:

- Four identical Parallel Processors (PPs).
- A Master Processor (MP) with RISC architecture.
- 50 Kbytes of SRAM cache.
- A Crossbar Switching Network.



TC = Transfer Controller
 PP0-3 = Parallel Processors
 MP = Master Processor
 VC = Video Controller
 FPU = Floating-point Unit

L = Local port
 I = Instruction port
 G = Global port
 C/D = Cache/data port
 TAP = Test access port
 OCR = On-chip register port

- A Transfer Controller (TC).
- Two Video Frame Controllers.

Each PP is an advanced 32-bit DSP with special features to improve the performance in image processing algorithms. The four PPs provide much of the TMS320C80's computational power. The conception of the data unit is one of the most remarkable features of the parallel processors. Its flexibility can manage and process several data in only one clock cycle. It offers the possibility of splitting the data unit to process one 32-bit word, two 16-bit integers or four bytes simultaneously. However, to obtain a good performance of the PPs, they have to be programmed, or at least part of the algorithms, in Assembly Code.

The MP is a 32-bit pipelined processor with an integral IEEE-754 floating-point unit. The floating-point unit, also pipelined, is able to start a new floating-point instruction on each clock cycle.

The 50 Kbytes of cache memory is organized in 25 banks of 2 Kbytes, and allows simultaneous accesses from the processors. 32 of the 50 Kbytes of cache are shared among the five processors, allowing a wide variety of configurations for parallel algorithms.

A high-speed crossbar network tightly connects cache memory and processors. This crossbar allows five instruction fetches and ten parallel data accesses per cycle.

An integrated transfer controller manages memory transfers in the cache memory, even from and to off-chip memory. Moving blocks of memory from off-chip memory to cache provides a much faster data processing. Besides, this transfer controller is image processing-oriented and it can be programmed to move rectangular regions of images, taking into account image coordinates and linear addressing.

3. Tracking

Tracking is a basic process in visual systems whose main goal is to keep an object of interest localized and fixated, pursuing it when the object is moving in the field of view [Crow95]. Tracking also tries to keep the object of interest in a certain and smaller region of the image called *fovea*. According to its simplicity, tracking process does not manage situations such as occlusions or very fast movements. Those cases must be managed by higher level processes.

There are two main behaviors that are desirable in a basic tracking module of a visual system: first, moving the camera pursuing the object and maintaining it centered in the fovea and second, detecting when the object has been lost.

3.1 The relocatable fovea

When continuous real-time processing rate is required, the amount of data that can be processed in bounded time is limited. Due to this reason, it is necessary to reduce incoming data to process, for example windowing them [Crow95]. Foveal formats [Pane95], among others, have been proposed. These formats focus its attention on the central part of the image, considering that part as the region of the image where interesting information is located. Therefore, whether this camera is mounted on a robotic-head, the motors have to be fast enough to follow the object and keep it in the central area of the image.

Pursuing and keeping the object of interest on the center of the image depends basically on two factors: time response of image processing algorithms and time response of electromechanical components. However, in most of cases, electromechanical time response is higher than the time a processor takes to compute an image. Because of that, tracking an object is constrained by the second factor. In others words, the speed of the object to follow could not be faster than

the reaction time of the motors. This fact does not take advantage of the higher speed of the processors. To overcome this, a relocatable fovea schema is proposed. In this schema the fovea is not always on the center of the image but it can move over the image, allowing to track the object even when this is not in the center of the image but in the border. This technique provides an extra time to the motors to react and adapt its velocity to the object.

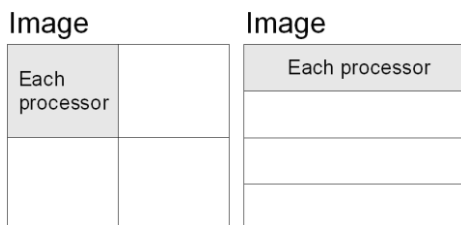
3.2 A SAD algorithm for C80

A SAD (Sum of Absolute Differences) algorithm is used to determine the location of the searched pattern on images. It computes the absolute difference between the pattern's pixels and a same size subimage's pixels on an image. Then, all these absolute differences are added giving a simple result. Mathematically:

$$I(x, y) = \min_{x,y} \sum_{i=0}^{size_x} \sum_{j=0}^{size_y} abs(image(x+i, y+j) - pattern(i, j))$$

The position in the image where this algorithm returns the lowest value is considered the best match, i. e, the object has been found at this point.

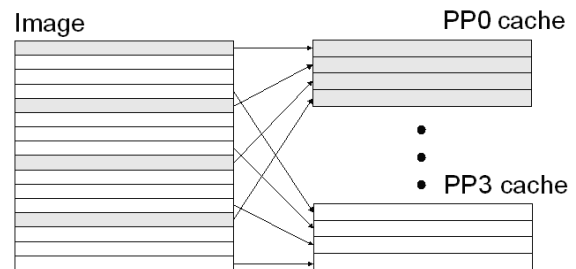
To parallelize this SAD algorithm, the first decision to take is how to divide its code and data among the available processors. The division will depend very much on the hardware architecture of the system. Every processor must be working as long as possible. In this case, a TMS320C80 is used and the image will be the fovea. Having four identical processors the problem is how to split and send parts of the image to each PP. Conventional ways to divide the image by four could be:



When the pattern is outer the edge of the divided image, a process of overlapping must be done with the adjacent processors. Besides, this data division does not assure that every processor operates always with the same amount of data, it means that not all the processors will take the same processing time. These kind of facts break continuity and homogeneity to the algorithm, and what is even worse, leave some processors idle for some time. To avoid all these problems, a better way to divide the image must be designed.

The proposed algorithm divides the image by four, as previous schemas, but sending to each PP's cache banks only one row each four. In other words, PP0 receives in cache the rows 0,4,8,12,16th ..., PP1 receives in cache the rows 1,5,9,13,17th ... and so on.

Besides this, each PP has a whole copy of the pattern in cache as well. Then, the pattern is shifted over the quarter of image as if it was a normal correlation algorithm. The difference comes from the lines of the



pattern that the algorithm takes in every row. Let us suppose that PP0 is starting its operation. It has a quarter of image which is composed actually by the 0,4,8,12th rows of the original image. From now on, we will call this quarter of image *quarter*. At the beginning, both pattern's and quarter's left-upper coordinates match up. Row 0's pixels of pattern and quarter are computed together, nevertheless rows 4,8,12... of pattern are computed with rows 1,2,3,... of quarter. In a second stage, when pattern's left-upper coordinates aligns with coordinates (x=0,y=1) of quarter, the

correspondence between rows would be: rows 3,7,11,... of pattern with rows 1,2,3,... of the quarter.

As the pattern is shifted horizontally along the rows, partial results are obtained by the PPs. In this manner, overlapping does not occur, and each PP works exactly the same amount of time and without being inactive during any time. Once finished one row, the four partial results are combined into only one by the MP. This combination only means to sum four vectors of results element by element, since these are actually partial results.

This algorithm uses several C80's powerful features [Prec]. The fovea and the pattern are loaded completely in cache memory speeding up considerably the performance. The size of this cache allows to store grayscale images of 128x128 pixels and four copies of the pattern of 32x32 pixels. Having several cache banks each PP can access to its part of image without causing idle cycles each other.



4. Experiments

C80 performs a correlation of a pattern of 24x24 pixels over a fovea of 80x80 pixels in 21.7 msec. which is less than 40 msec. There are still 18.3 milliseconds left which allows integrating several improvements to the algorithm. In the image the system

tracks a person who is moving in a room, the numbers provides a temporal reference.

Conclusions

An Active Vision System for real-time tracking of objects is presented. This system has been composed based on off the shelf components.

For low level processing a C80 DSP has been used, providing a good performance for primitive real-time tasks which are essential for higher level vision systems.

The proposed algorithm makes use of a singular way to divide the image, which allows all PPs work without idle time and with an optimum charge balance. It is also suitable for other algorithms with a similar kernel.

References

- [Aloi87] Aloimonos, J. Weiss, I. and Bandyopadhyay, A. "Active Vision" International Journal of Computer Vision,1(4):333-356.
- [Bajc88] Bajcsy R, Active Perception, Proceedings of the IEEE 76(8):996-1005
- [Crow95] Crowley J. L., Christensen H. I. (eds.), Vision as Process. Springer-Verlag, Berlin, 1995.
- [Pahl93] Kourosh Pahlavan, "Active Robot Vision and Primary Ocular Proceses",1993-CVAP.
- [Pane95] F. Panerai, C. Capurro, G. Sandini, "Space variant vision for an active camera mount", TR 1/95, LIRA-lab - DIST University of Genova, Italy, 1995.
- [Prec96] "Precision MX Video Engine. Technical Reference". Precision Digital Image Corporation. 1996
- [Seel96] Ulf M. Cahn von Seelen, Brian C. Madden, "A Modular Architecture For An Active Vision System Using Off-The-Shelf Components" GRASP Laboratory. Univ. of Pennsylvania. 1996