MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

HERMIA: AN HETEROGENEOUS AND RECONFIGURABLE MACHINE FOR IMAGE ANALYSIS ^

Gaetano Gerardi(*), Franco Chiavetta(+), Vito Di Gesù(+), Domenico Tegolo(+)

(*) Istituto di Fisica, Università di Palermo
Via Archirafi 36, 90123 Palermo, ITALY
(+) Dipartimento di Matematica ed Applicazioni, Università di Palermo
Via Archirafi 34, 90123 Palermo, ITALY

ABSTRACT

In this paper is described the general architecture of an Heterogeneous and Reconfigurable Machine for Image Analysis (HERMIA); the first prototype of the system has been developed at the University of Palermo. Conventional hardware has been used in order to emulate the machine and evaluate the system performance. Preliminary results are presented and discussed.

INTRODUCTION

The image analysis computation paradigm follows four phases: low level processing, (LLP), (examples are filtering, digital transformations,...), intermediate level processing, (ILP), (examples are segmentation and feature extraction), high level processing, (HLP), (examples are classification, structural description) and interpretative processing, (IP), (examples are semantic description, physical model extraction).

Each of the these phases is characterized by different levels of abstraction. For example LLP uses often pixel and local operators, HLP is performed by using global operators acting, for example, in a feature space previously selected, IP needs large database and knowledge base (organized perhaps in semantic networks) in order to describe the expertise of the users.

The full integration of all phases in a unique system is still unsolved. The approach here described is based on the principle of specialization and cooperation between workers. The image analysis is carried out in a pipe fashion; each steps is performed by a dedicated class of processors (heterogeneity). The pipe is reconfigurable as well as each element of the pipe. Reconfigurability seems to be one of the key-features in the solution of vision problems [1,2].

The first prototype of the HERMIA-machine has been developed at the University of Palermo inside of a research project supported by the Italian Council of Research. Conventional hardware has been used to realize the first prototype of HERMIA in order to emulate the machine and evaluate the system performance. The following sections describe: the general architecture proposed, the hardware implementation, the software environment, the evaluation results and the final remarks.

PROPOSED ARCHITECTURE

Overall architecture. In Figure 2 the overall architecture is shown.The system consists of four main modules in order to perform different levels of the analysis. The LLP module is dedicated to low level computation; the ILP module performs the intermediate computation; the H module performs both high and interpretative analysis; the last one is also responsible for the software management (operating system tasks, editing, compilation and loading) and control (I/O interrupt and tasks synchronization) of the whole system. Finally the active memory (AM) module handles the I/O of the image data. Image data can be accessed separately by all the functional modules by means of a run-time memory bank switch.

The modules LLP, ILP and AM are fully connected by high speed serial links (20Mbit/sec). This architecture allows a fast communication of data among these modules; Their connection with the host, performed via an high speed parallel bus is confined for loading programs and I/O of small data sets. This corresponds to a conceptual separation between the LLP and ILP modules, which need to exchange often data in order to solve heavy numerical problems, and the H module, which perform mainly non-numerical analysis.

HARDWARE IMPLEMENTATION

LLP module. It consists of a set of processors dedicated to low level operations. At the present time it consists of an INMOS B009, which includes 4 IMS A100 signal processors (DSP) controlled by a TRAN B411. The 4 IMS A100's can be used to implement 128 tap FIR filters, convolvers or correlators, on 16 bit data, with 16 bits coefficients at rates up to 2.5 Msamples/second. This module allows, for example, the development of an efficient FFT algorithm (0.1sec for an array of 512x512 elements). Future developments foresee the implementation of this module with more powerful and image oriented DSP. For example A110 DSP allows to perform 2D operators at the rate of 20Msample/sec with 8bits data. The size of the data and coefficients can be determined by hardware reconfiguration without speed penalty. Moreover a postprocessor unit is included in the chip; this feature allows to perform, in pipeline, for example: look-up table or statistics or scaling of the image data [3], avoiding the use of separate units with exchange-data overhead.

ILP module. The module will consists of an array of 16 transputers INMOS B411 controlled by an INMOS B417. They are fitted in two IMSB008 housed in a PC/AT. The topology of the interconnection network can be reconfigured (linear array, mesh, cube, hypercube, ring and pyramids) by changing the configuration of the four transputer-links with a programmable 32 way crossbar switch. This feature allows to optimize the module-topology for the selected set of algorithms.

^ Work supported by the Italian Council of Research under contract n° 89.00216.12

AM module. The main purpose of the module is to assign dynamically, under the task-control, to each module the image memory bank as required. At the present time image-data are exchanged by the described busses; the selection of sub-windows is performed interactively using the performance of a graphics system (1280x1024x8). It is realized by an INMOS B419, which consists of an IMS T800 32 bit transputer and IMS G300 colour video graphics controller. Two Mbytes of four cycle DRAM provides a general purpose store sufficient to run application such as X-window system.

H module. In the present implementation the host consists of a PC 386, with 4Mbyte of RAM and 320Mbyte hard-disk memory. This configuration allows to perform the above described control-tasks and high level vision analysis.

SOFTWARE ENVIRONMENT

The system software. It is the TDS2, which is resident on the host computer. It provides the management of the interfaces between peripherals and the modules LLP and ILP, as well as the editing and the compiling of OCCAM programs. TDS2-programs may be also developed on ROM in order to realize special purpose functions.

Programming language. Multiprocessors machines need for concurrent programming languages, which allow users to express and handle concurrency and parallel statements in natural way as in standard high level programming languages. Moreover tools for the debugging phase must be provided to control the general status of the computation in run time. Graphics interactive environment can be very useful, given the complexity of the hardware topologies. At the present time Modula-2 under DOS has been considered. It fit most of the general requirements above listed. For example it includes source level interactive debugger, graphical user interface library. Moreover Modula-2 allows dynamic network configuration and processing spawning. This feature is basic in reconfigurable systems.

The pictorial language. Pictorial languages have been developed to handle high level vision problems [4, 5,6,7]. They allow to define pictorial objects and operations on them, as well icon-guided navigation throughout the pictorial data bases.

On the other hand the need exists for developing medium level languages to design and to implement low and medium level vision algorithms on multiprocessors machines.

A pictorial C language (PICL), has been designed to develop vision algorithms on the HERMIA machines [8,9,10]. Its first release has been included in the software environment of the pyramid machine PAPIA [11,12]. It is an extension of the C language, where parallel instructions (where...otherwise, pwhile, pdo...puntil, select, parallel assigment) and image-data-structures (IMA, PYRA, CUBE, FIELD) are included.

PICL allows the execution of asynchronous processes. Concurrence is handled by means of an open/close contexts mechanism. Informally the contexts are subsets of processors of HERMIA. The open (close) statement enables (disables) the processors to whom it is applied. The execution of a list of processes may be sketched as follows:

"given a set $\{P_i\}$ of processes, assign them to a set $\{C_i\}$ of contexts, two processes, P_1 and P_2, may be executed asynchronously only if the corresponding contexts, C_1 and C_2, are set-disjointed and the open/close statements of C_1 and C_2 are not interleaved in the algorithm".

For example the first vision level of an image analysis procedure requires, often, for local computations performed on disjointed, or partially shared, sub-scenes [13]. The results, so obtained, may then be used at the higher levels of the computation phases. For this purpose sets of processors (disjointed or partially shared) are assigned to each sub-scene to solve concurrent local tasks.

Higher performance is expected, whenever the best match between contexts and processes is reached and the hardware fits with the data structure. From this point of view contexts are, also, treated and defined as data, they may be assigned to a constant-context, or to an expression of contexts. They may be dynamically allocated and disallocated.

SYSTEM EVALUATION

The performance evaluation of the HERMIA machine has not been yet completed; in the following some preliminary bench-mark results are reported. They are based on low and intermediate level algorithms (Look_up table, tresholding, histogramming, mean, Sobel filter, convolution, equalization, distance transform, component labeling). The actual hardware configuration consists of a IMSB008 in which are plugged nine T800; one of them is the master of a network of eight transputers. The architecture topology used is a bidirectional linear array. The images are horizontally partitioned inside the master. It sends, in parallel, the first part of the image to the head of the pipe and the second part to the tail. Therefore, the architecture topology can be phisically considered also as a ring.

Network performances are monitored by using the following times, measured with a precision of 64 microsecond:

T1 = sending image start time.
T2 = sending image end time.
T3 = receiving image start time.
T4 = receiving image end time.

The master never participates to the algorithm execution, while it is responsible for the detection of T1-T4 and the evaluation of the overall performance.

The algorithm execution, in each transputer starts at the same time; when each processor has received its own part of image. Each transputer in the network monitors the starting and the end time of the algorithm execution (Ts and Te). These information are sent to and collected by the master which calculates the computation time (ct(i)=Te-Ts) for each processor "i". In order to have a reference, for performance parameters evaluation, Serial Computation Time (SCT) is defined as the time spent by a single transputer (the master) to execute an algorithm in serial mode. In the following the performances parameters definitions are given:

Execution time	Et = T4-T1;
Computation time	Ct = T3-T2;
Sum of computation times	Sct= ct(1)+ct(2)+...+ct(n);
Overhead Time	Ot = (T4-T3)+(T2-T1)=Et-Ct;
Overhead Ratio	OR = ot/Et;
Utilization	Ut = Sct/(n*Et);
Speed-up	Su = SCT/Et;
Efficiency	Ef = Su/n= SCT/(n*Et);
Redundancy	Ri = Sct/SCT;
Computational speed-up	CSu= SCT/Ct;

Where "n" is the number of transputers and the Overhead time (Ot) is the time spent by the network to receive and return images. It does not depend on the transputer number in the network; in fact each trasnsputer

has two, parallel active, channels: while one receive a new partition, the other one transfers the previous partition to the neighbouring processor. This kind of parallel operation is possible only if two push-pull are declared. The experiments have been carried out with input images of size 64x64x8. The computation is made by using 32bits words.

In Figures 1a,b,c the computation time, Tc, versus the number of transputer (Ntran) is shown. The figures are referred to point operators (LUT and TRESH), to local operators (MEAN and SOBEL) and to global operators (EQUA and LABEL) respectively. The algorithms MEAN and SOBEL perform local operations by using a 3x3 kernel.

The experiments show that Ct decreases with Ntrans, as expected; however the asymptote is higher for local and global operators, because of the communications between processors.

Note that for Ntrans>8 the Ct variation is very small (~11%).

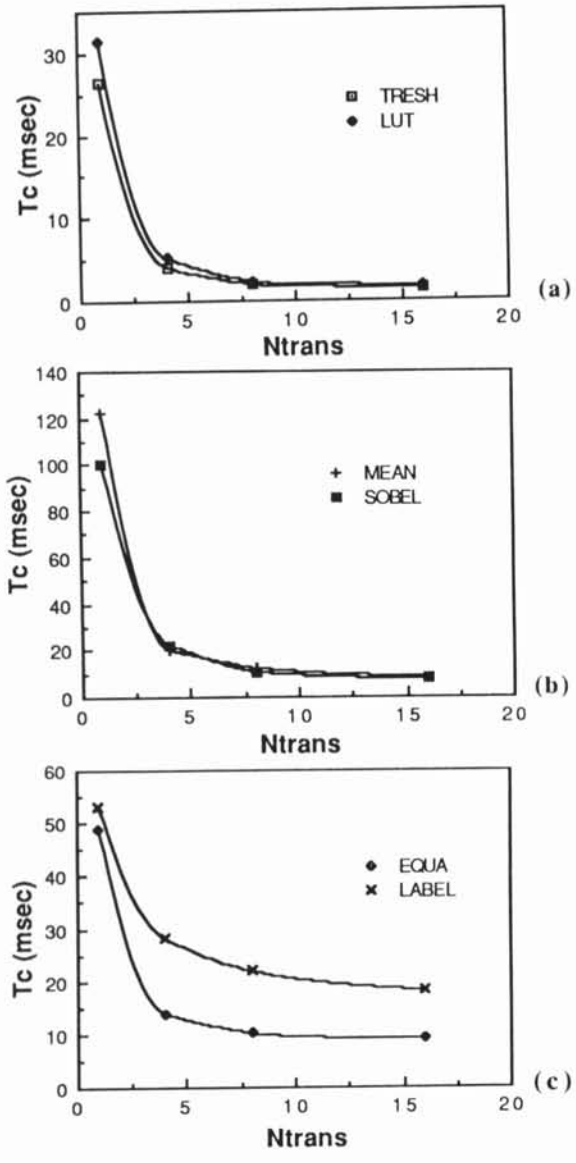

Figure 1. a) point operators; b) local operators; c) global operators.

FINAL REMARKS

The paper show a new multiprocessor machine, based on dedicated and general purpose reconfigurable processors. Heterogeneity is useful in order to perform repetitive visual-functions. Reconfigurability in real-time is necessary in order to tune the architecture with the algorithm. HERMIA-machine will be tested on real applications in the near future.

REFERENCES

[1] F.J.Weil, L.H.Jamieson and E.J.Delp, "DISC: A Method for Dynamic Intelligent Scheduling and Control of Reconfigurable Parallel Architectures", Purdue Univ. TR-EE-89-14, 1989.

[2] F.J.Weil, E.J.Delp, "Dynamic Intelligent Scheduling and Control of Reconfigurable Parallel Architectures for Computer Vision / Image Processing", Proc. 10th Int.Conf. on Pattern Recognition, IEEE Comp.Society Press, Atlantic City, 1990.

[3] "The Digital Signal Processing Databook", INMOS, SGS-Thomson, 1989.

[4] K.Preston, Jr., "Progress in Image Processing Languages", in Computing Structures for Image Processing, edit.M.J.B.Duff, Academic Press, London, pp.195-211, 1983.

[5] S.Levialdi, A.Maggiolo-Schettini, M.Napoli, G.Uccella, "PIXAL: a High Level Language for Image Processing", in Real Time/Parallel Computing, M.Onoe, K.Preston Jr., A.Rosenfeld, edits., Plenum Press, pp.131-143, 1981.

[6] Man-Chi Pong, "A Graphical Language for Concurrent Programming", IEEE Workshop on Visual Languages, Dallas, 1986.

[7] R.A.Finkel, J.L.Bentley,"Quadtree: a Data Structure for Retrieval on Composite Keys", Acta Informatica, Viol.4, 1974.

[8] V.Di Gesù, "A High Level Language for Pyramidal Image Processing", in Pyramidal Systems for Computer Vision, NATO ASI Series F, Vol.25, pp.329-339, 1986.

[9] D.Tegolo, V.Di Gesù, and B.Lenzitti, "Definition and Properties of a New Pictorial Data Type", in proc. 1989 IEEE Workshop on Visual Languages, IEEE Comp.Society Press, pp.117, 1989.

[10] V.Di Gesù, "Pictorial Languages: concepts and data structures, invited paper, Special Issue JIETE on Pattern Recognition, in press.

[11] V.Cantoni, M.Ferretti, S.Levialdi, F.Maloberti, "A Pyramid Project Using Integrated Technology" in 'Integrated Technology for Parallel Image Processing', S.Levialdi Ed., Academic press , New York, 1984.

[12] V.Cantoni, L.Carrioli, O.Catalano, L.Cinque, V.Di Gesu', M.Ferretti, G.Gerardi, S.Levialdi, R.Lombardi, A.Machi', R.Stefanelli, "The PAPIA Image Analysis System", SPIE International Symposium on Image Processing, Cannes, 1985.

[13] Man-Chi Pong, "A Graphical Language for Concurrent Programming", IEEE Workshop on Visual Languages, Dallas, 1986.

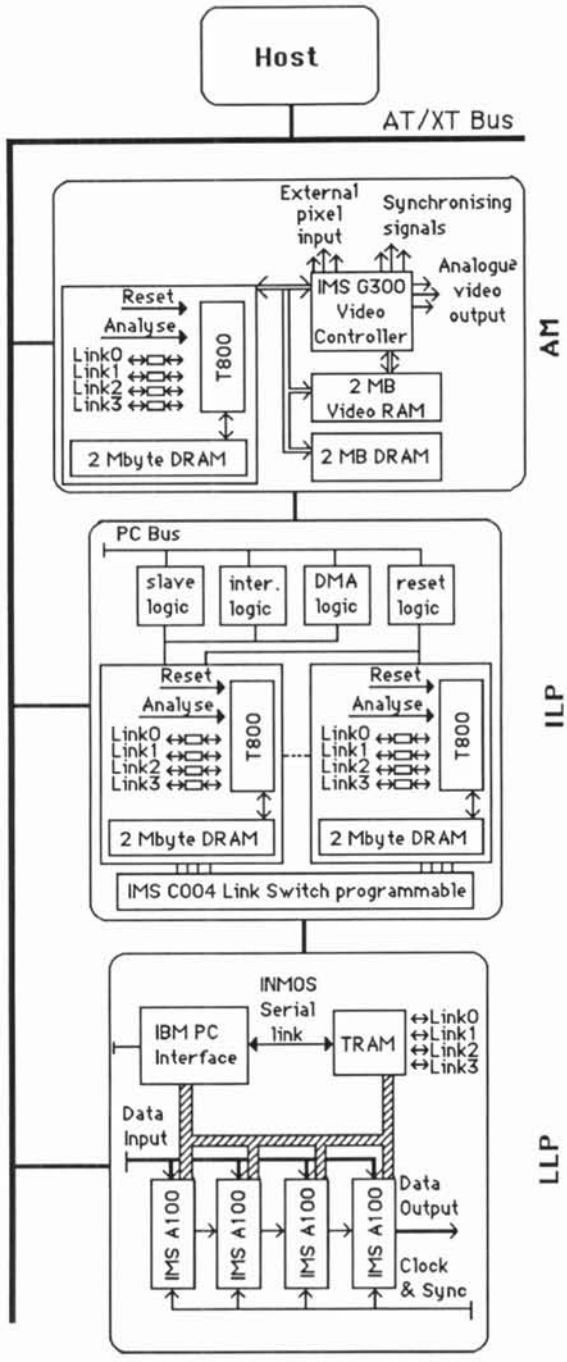

Figure 2. Architecture proposed.