IAPR Workshop on CV - Special Hardware and Industrial Applications OCT.12-14, 1988, Tokyo

IMPLEMENTATION AND EVALUATION OF ALGORITHMS FOR IMAGE PROCESSING BY MEANS OF TRANSPUTER NETWORK

M. AKIL

Laboratoire Intelligence Artificielle et Analyses d'Images
GROUPE ESIEE BP99 - 2, Bld Blaise Pascal - 93162 NOISY-LE-GRAND Cedex France

ABSTRACT

The problem investigated in this paper is to find a programming method for a Transputer type, processor network and to evaluate its performance on various algorithms for image processing.

The original feature of our study is based upon the design of a modular architecture (acquisition/display module, Transputer module) and the definition of a programming method that is independant of the processor network topology. The architecture allows to connect several 4 transputers modules.

The programming method defines 2 models : **communication and processing** and uses a **communication scheme** that ensures the portability of the application on the target configuration. The communication model is enhanced to be **'full model'** so that an algorithm can be implemented on any physical configuration.

Our method has been tested on a 4 transputer network, it can be extended to a higher number. The measured response times show the efficiency of such a network and let hope to reach real time processing by adding transputer modules. The efficiency is increased by using parallelism between communication and processing, and managing consistently both of them to reduce the loss of time due to synchronization.

INTRODUCTION

The amount of information in a picture as well as the type of problem to solve, lead to design and make efficient specialized architectures that use various types of parallelism.

The parallelism (on pictures, operators, ect...) may be executed in different modes : SIMD (Single Instruction Multiple Data) or MIMD (Multiple Instruction Multiple Data). Parallelism is implemented in such machines to achieve specific processing : picture improvement, filtering, etc... : «Low level processing». Other architectures, such as processors network, automata network using an image representation, allow from a knowledge base the understanding of the picture to be analysed : «High level processing». So it appears useful to design a vision machine including the various steps of the process leading to pattern recognition.

Also, parallelism may use either the **algorithm properties** : the elementary tasks are sequentially applied to different parts of the image or the **data structure** : the same algorithm proceeds different parts of the picture or is simultaneously applied to different regions.

Moreover the computing power of these machines is increased by the use of VLSI (Very Large Scale Integration) programmable parallelism-oriented processors such as transputers.

This paper describes an architecture for image processing including a module for acquisition/visualisation, a modular structure based upon transputers. We develop then a programming method for transputer networks and give some performance measurement for several image processing algorithms.

ARCHITECTURE

In a multi-processor structure, consisting of standard processors connected to the image memory, one of the problems is to ensure a sufficient data rate between processors and memory for the intended application. Most often the bus is the data bottleneck for that type of structure. Several solutions may come under consideration : high data rate bus, ring type bus, etc... , this will improve the data flow but will not avoid saturation.

Our approch consists of defining an asynchronous MIMD multiprocessor architecture wich uses transputer type (from INMOS).

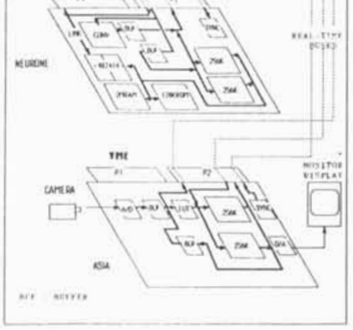
The TRANSPUTER is a 32 bits microprocessor (10/15 mips) including RAM and process management. Moreover it has 2 important characteristics : **4 bidirectionnal serial communication ports** (channels) from 5 to 20 Mbits/second and **parallel programming easiness** thanks to OCCAM high level language, intended for that purpose.

These links or channels allow to connect several transputers without any common bus and to elaborate different types of configuration.

Overall Presentation : The system includes a host computer and 2 modules. One module : **acquisition/Visualisation (ASIA)** is used for the following functions : Camera interfacing, A/D 8 bits and D/A conversion and video timings control. It includes a LUT (look-up table) on the input and 2 memory planes.

The «NEURONE» coprocessor module intended for image processing allows real time vector edge detection, real time 3*3 Sobel convolution in 4 directions (South-North, ...). It includes 4 Transputer, 2 memory planes (512*512) and also system memory (2Mb RAM and 128Kb EPROM). S

The host computer is intended to generate the controls and then analyse the results. The schema hereafter presents the connexion between the 2 modules.



System Operation : The ASIA module digitalises and displays the image, it manages the video signal coming from the camera and sends them to the monitor. After being digitalised by means of an A/D converter, the image is translated through the LUT and then stored in one of the 2 memory planes of ASIA. The host computer can modify the LUT, decide to send orders for image acquisition or display and manage the video Timing on ASIA. Also it can transfer an image plane into another trough the LUT. This allows us to avoid a new acquisition if the LUT is not properly adapted.

The NEURONE coprocessor module is intended for real time operations (convolution, edge detection). It extracts from previous results or from the original image some specific characteristics by means of an appropriate processing and then generates the result for the host. The edge are stored as vectors with amplitude and direction. The image is stored in 1 of the 2 planes of this module. It is noticeable that only the master transputer may access the memories and decide to receive the image pixels from the camera either directly or through the convolution circuit. It communicates with the host and with the 3 slave transputers by means of the serial channels. One of the functions of the master is to distribute the information to the slaves. One can add the appropriate number of modules for a given application, the links on the transputers are available to allow us to define various architecture topologies. 1 to 4 NEURONE modules may be connected to the ASIA module. The image from ASIA is simultaneously generated to the NEURONE modules. In this way adding modules improves the computing power. The efficiency of such a structure will also depend on the way of implementation of the algorithm on the network.

ALGORITHMS FOR IMAGE PROCESSING AND THEIR IMPLEMENTATION

OCCAM language : OCCAM comes out from CSP (Hoare 78) and is closely related to it. In OCCAM the channels are explicitely defined by a port common name. A specific symbol indicates the direction of the exchange : «!» for transmission and «?» for reception. OCCAM can handle and build up processes (the elementary process consists of a single instruction, an empty process is SKIP) by means of constructors. SEQ makes up a set of sequential processes, PAR allows parallel execution of processes and ALT defines a pseudo non determinism by execution of 1 process out of the ready ones at a given time. PRI PAR defines a parallel execution with priority given to the first process.

Let P1 and P2 be 2 processes executed on one transputer, P1 and P2 are then logical processes and will communicate through a logical channel. If P1 and P2 are executed respectively on T1 and T2 then P1 and P2 will represent physical processes and will communicate via a physical channel.

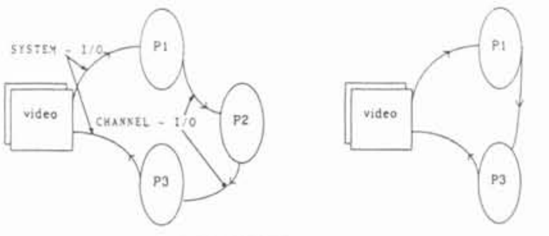
Programming method : The parallelism induced by the transputers (process management, communication between processes) leads to a specific functional organisation for the application. One of the aims of our method consists of isolating, for a given program on the one hand the communication and process management specific part and on the other what is specifically involved in the processing.

The other purpose will be not to consider the proposed architecture (ASIA/NEURONE) as a target but as a mean to evaluate the algorithms to be implemented. This will allow us to ensure a proper implementation of any algorithm on various network topologies.

These 2 aims will be reached by the introduction of 2 concepts :

a) the communication model is a set of processes intended to communicate with each other according to one or severals protocols. Each process has one or several I/O (Input/output). These ones are classified into channel I/O (relation between 2 functions for transmission or reception) and system I/O allowing communication with external components. The first purpose of that type of process is to receive and transmit information. A processing module will have to be integrated between the reception part and the transmission part of a process in a communication model.

b) Also, a communication scheme is the result of the connection one with another and also with outside components of a number of processes taken from a communication model. The hereafter schema presents a communication scheme.



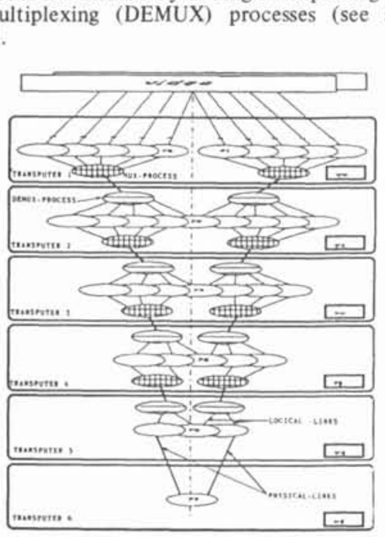
COMMUNICATION-SCHEME

In this schema the associated protocol may be the processing of an image line.

P1 will read and transmit one line after another a part of the image, while P2 if it is not assigned to a processing model will simply transmit lines. P3 will store sequentially the lines received to the video memory. P2 may be assigned to a processing module.

The problem for the implementation of a communication scheme on a given network is the existence of a theoretically unlimited number of logical links between communication models. The implementation may then be in fact impossible because of the finite number of physical links in the network.

The given solution is then to make a «full communication model» that allows to implement any communication scheme on a given architecture. This characteristic of the model is obtained by adding multiplexing (MUX) and demultiplexing (DEMUX) processes (see schema hereafter).



" FULL COMMUNICATION MODEL "

In this way if we organise any algorithm as the association of a communication model and a processing model then the implementation onto NEURONE (or any geometrical topology obtained by configuration of NEURONE modules) will be made possible.

Algorithm Implementation : The first step is to study the algorithm by describing the necessary data types for the processing module. The choice criteria for the algorithm will be implementation easiness and performance constraints. The size constraint (i.e the available memory space on the transputer) must be taken into account.

Data types depend on the algorithm : local algorithms (for example processing one point independantly of the others as in the threshold determination) ; neighbourhood algorithms (as convolution wich processes one point in relation with its neighbours) ; line algorithms (global processing of an image line : panning) and overall algorithms (process the total picture example : histogram).

The second step is to evaluate the necessary storage size for the initial data, the intermediary result and the program.

The choice of the communication model is the third step, it will also include choices for partitioning the algorithm, image distribution, and a mode for the data exchange. Dividing the algorithm into modules that will be distributed on several transputers has the drawback of increasing the data flow and subsequently the cost of synchronisation between the transputers. Most often the choosen solution is not to divide the algorithm.

The choice of the way to represent the image will be done according to the type of data needed by the algorithm, taking into account the amount of available memory on a single transputer. In that way a horizontal distribution (image processing on a one line block basis), although it requires more memory, leads to a noticeable speed increase. Should this way be impossible, a vertical repartition (line processing on a points block basis) will be used. Implementing an algorithm consists then of gathering into a communication scheme, the various written processes of the model and assigning it to the network (declaration of the channels).

Algorithm optimisation : This phase consists of :

1-use parallelism between communication and processing i.e :

```
do in parallel for i ranging from 1 to N
    receive (data (i+1) )
    proceed (data (i), results (i)
    send (results (i-1) )
```

2-give communications the priority by :

```
do in parallel with priority for i from 1 to N
    do in parallel
        receive (data (i+1) )
        send (results (i-1) )
    proceed (data (i), results (i) )
```

3-optimise the communications by simplifying the protocols and reducing their number by increasing their density. This optimisation reduces the added cost for synchronisation.

4-use specific resources from OCCAM.

Example of an implemented algorithm : Threshold processing consists of changing the intensity for every pixel in the picture.

The communication model is based upon a single protocol. The data type is line based. The model is made up of processes.

P1 gets the image one line at a time and transmits these data to the networks that are available. P2 stores the line into the memory after getting them from the network (channel I/O). P4 is an equivalent for P3, assigned to the process module. This module is a procedure with a full image line as input / output argument. P5 makes the multiplexing (Channel I/O). P6 makes the demultiplexing. These last 2 processes ensure a full communication model.

The implementation of others algorithms has been investigated according to the method described above.

Performance evaluation : We shall consider here an algorithm executed with a network (P identical processors) in a tp elapsed time while the sequential execution on a single processor would be t1.

We give hereafter a list of measurements for several algorithms with t1 and t4 (4 transputers network).

Function	Image type	t1 (ms)	t4 (ms)
Threshold	256*256	394	83
Panning	256*256	61	52
Erosion	120*256	700	168
Histogram	256*256	1362	211
Histogram equalization	256*256	1369	220
Image comparison	512*512	2231	513
Convolution	512*512	5119	869

These figures indicate an important acceleration ratio between 1 and 4 processors. The efficiency is accordingly better as it comes from reducing the cost of parallelism management (synchronization, communication, etc...) and using the whole of the resources of the OCCAM language. This optimization has been done by the programmer leading to a large programming time.

Conclusion : The architecture described above is a step towards the design of a structure that includes both «low level» and «high level» process.

We have checked the efficiency of the proposed architecture and the use of transputer for global operations.

The definition of a programming method has allowed a structured programming approach for the transputer network and has ensured an easy portability onto any network geometrical topology.

These works have been realized with the contribution of **M. G. BOUCOURT (VMES Domaine du Rouret - 30380 St CHRISTOL-LES-ALES - FRANCE)** for the Hardware parts ; **MM. C. BENOIT and B. FALL (Groupe ESIEE - FRANCE)** for the Software parts.

Références : - 7th OCCAM users group & International Workshop on Parallel programming of transputer based machines Grenoble LGI-IMAG sept 14-16 1987
- OCCAM 2 Reference manual INMOS Ltd ED
- Image Processing System Architecture ED. by J. KITTLER and Michael J.R DUFF : R.S. Press LTD - April 1986
- ARRAY Architectures For Iconic and SYMBOLIC PROCESSING by T.J. FOUNTAIN - IEEE 1988
- Digital Picture Processing by A. Rosenfeld and A.C. Kak - Vol 1 and 2 - Sd Edition