

SLAC - PUB - 4059

August 1986

M

AN OVERVIEW ON PYRAMID MACHINES
FOR IMAGE PROCESSING*

VITO DI GESU'

Dip.di Matematica e Applicazioni, Univ.di Palermo, ITALY

and

*Computation Research Group, Stanford Linear Accelerator Center
Stanford University, Stanford, California, 94305*

ABSTRACT

This paper is intended to review the existing hierarchical multiprocessor machines dedicated to Image Processing (IP). Some general design concepts are given and compared in terms of hardware and software complexity.

Invited lecture given at the 2nd International Conference
on Advances in Pattern Recognition and Digital Techniques,
Indian Statistical Institute
Calcutta, New Delhi, India, January 6-9, 1986

* Work partially supported by the Department of Energy, contract DE-A03-76SF00515.

I. INTRODUCTION

In Image Processing (IP) area the volume of the data (the number of pixels can vary from 64x64 in scintigraphic images, to 2340x2340 in four channels for LANDSAT images), and the evolution of computational demands (image understanding, robot vision) have grown in the last decade, ceaselessly.

In the past, the computational bottleneck has been mitigated by designing and building faster processors. However, physical constraints place limitations on this approach. VLSI technology makes possible new approaches based on the organization of several processing elements (PEs) in suitable architectures.

The PEs operate in a parallel fashion. They may send data to and receive data and instructions from neighboring PEs. Sometimes they are governed by a single control unit (CU), that broadcasts instructions and data and resolves conflicts and concurrence among the PEs. Different operational strategies allow the PEs to control the computation tasks by themselves. This last operational mode is more complex and requires very powerful PEs and fast interprocessor communication channels as the conflict and concurrency problems are more complex.

The cooperation of many PEs for solveing a single task or a complex set of tasks by distributing the problem to specialized PEs, seems to be the only way for increasing the speed for the computations required in IP. In fact several thousand of PEs may compute simultaneously, in an asynchronous way, billions of elementary instructions per second.

A complete image analysis system requires the following stages:

- a) PREPROCESSING. The main goal is to prepare the image for further automatic analysis. This stage requires point operations (thresholding, requantization, encoding,...), local operations (filtering, template-matching,...) and global operations (digital transformation, Fourier, Hadamard, Haar,...). All the pixels in the image are processed and the basic operations are simple.
- b) PRIMITIVE EXTRACTION. Salient features are extracted by means of an image segmentation (skeletonization, edge following,...) or on the

basis of the grey level statistics (histogramming, co-occurrences,...). Also in this stage all the pixels are processed.

- c) **SYMBOLIC DESCRIPTION.** The features are organized in more complex segments in order to realize a more compact description of the image and its components. This stage operates on less data with more complex structure. Artificial intelligence methods are, usually, required.
- d) **INTERPRETATION.** This is the final and has a more complex goal; it strongly depends on the previous stages. The analysis is performed by using pictorial symbols and knowledge related to the specific problem.

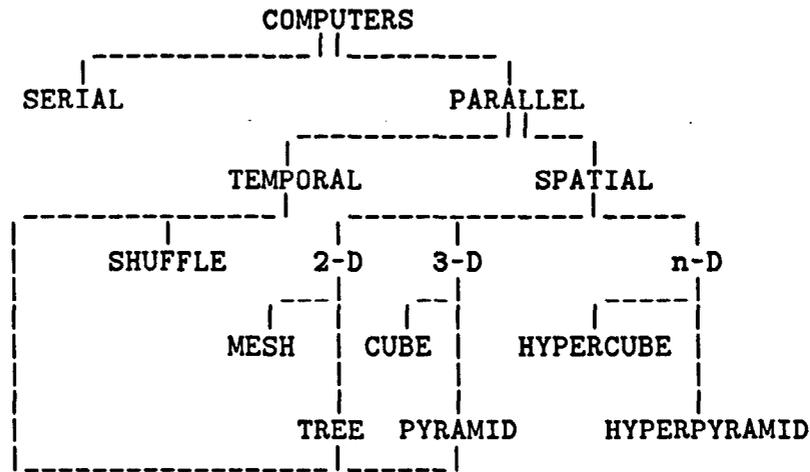


Fig.1 Classification tree of main existing computer architectures.

Stages a) and b) define the low level vision part of the image analysis process. They operate on the whole image with a large number of elementary operations. Stages c) and d) define the high level vision phase. They operate on complex data structures and sometimes use artificial intelligence techniques.

These four stages are very related and some feedback mechanism is usually included in order to reevaluate decisions taken in previous stages (backtracking).

In the whole analysis a large amount of the computation is consumed at the low level vision stages. Multiprocessor systems dedicated to that part of

the analysis seem to be effective. Many IP-machines have been designed and some of them realized in the last decade [1,2,3,4] for the solution of low level vision problems. They are usually integrated as part of an image processing system and some of them are oriented to a specific problem [1]. Others may be used for a wide class of applications [2]. In references [5,6] a good taxonomy study of the existing architectures dedicated to IP can be found.

In multiprocessor machines dedicated to IP, each PE may correspond to a single pixel of the image or to a set of them. The PEs may do point and/or local computation, depending on whether they operate on single pixel values or on neighboring values as well. Global computation is usually harder because it requires complex navigation of the data through the array of the PEs.

Several architectures have been proposed and Fig.1 shows a classification tree, leaves of which correspond to a specific class of machines. Among them the cellular machines as CLIP4 [2], in which each PE is interconnected in the plane with the four or eight neighbors, see Fig.2, and the pyramid machines [7,8], described extensively in section 2, seem to be, up to now, the most suitable for handling IP problems.

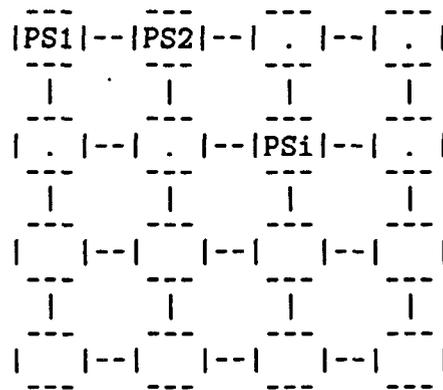


Fig.2 Cellular machine plane connection.

Section 2 is dedicated to a general description of the pyramid architecture and to the main motivations behind it. Section 3 describes the most relevant

research projects on pyramid machines. In section 4 a comparison between the machines discussed in this paper is made. Section 5 is devoted to concluding remarks.

II. BACKGROUND AND MOTIVATION

Informally, pyramid machines are cellular arrays of processors having hierarchical interconnections, based on the binary tree and the quadtree data structures. The concept that hierarchical organization of the data and subsequent actions on them fits many of the problems we are faced with, is quite old and has been applied to data base management [9], cluster analysis [10], image analysis [11], and multiresolution image representation [12,13]. The idea of layered pyramid machines dedicated to IP has been introduced by L. Uhr in 1972 [14] and formalized by S.L. Tanimoto and T.A. Pavlidis in 1975 [15].

Pyramid machines have been motivated by biological arguments that show a correspondence between its architecture and the mammalian visual pathway, starting from the retina and ending in the deepest layers of the visual cortex.

A multilayer array of processors is the natural way to store and process multiresolution images. The most recent ones are sequences of images

$$(A_0, A_1, \dots, A_L)$$

with corresponding resolution:

$$(2^L \times 2^L, 2^{L-1} \times 2^{L-1}, \dots, 2 \times 2, 1).$$

The divide and conquer rule is inherent to the pyramid and allows the reduction of computational cost. Many functions may be computed in $O(\log N)$ time instead of $O(N)$ where N is the linear size of the input image. Recursive computation fits well with the pyramid architecture. Vertical communications reduce the number of clock cycles for exchange of data ($\log N$ instead of N)

[16,17]. Pyramid architecture seems to be the natural ones for establishing links between information at different levels (e.g. from pixels to regions and vice versa) [18].

II.1 LINK CONFIGURATION

Pyramid systems have three main functional modules:

- the pyramid array;
- the control unit;
- the stage or active memory.

The control unit manages the flux of data and instructions from and to the pyramid array. The stage memory handles the I/O from and to the pyramid array; files of images are usually exchanged between the pyramid and the host computer. The stage memory design must provide for both fast I/O and data formatting in order to avoid to the extent possible the bottleneck arising from I/O operations.

In the following we show some architectural features of pyramid arrays. Three main link-strategies are described and discussed: the "basic pyramid", "the interleaved pyramid" and the "trees pyramid".

Formally the "basic pyramids" are a set of 3-tuples (i,j,k) , whose components may be considered to be coordinates (row i , column j , level k):

$$PM = \{(i,j,k) | 0 \leq k \leq L \text{ and } 0 \leq i \leq 2^k \text{ and } 0 \leq j < 2^k\}$$

the set PM is also known as the hierarchical domain of the pyramid with $L+1$ levels (see Fig.3). Each cell corresponds to a PE. Each PE has a set of neighbors, SN , directly linked to it. They are respectively the father the four sons and four or eight brothers, depending on the connections in the plane. The number of elements in SN therefore ranges from 9 to 13.

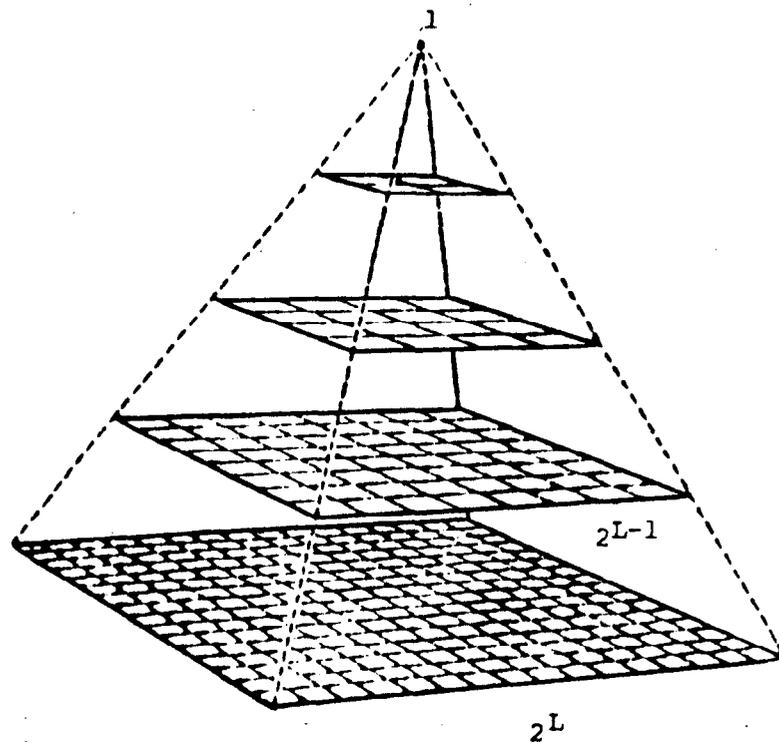


Fig.3 "Basic Pyramid" architecture.

Table 1 shows the coordinates of the SN of the PE located in the cell (i,j,k).

Table 1

F father	$(k-1, i \text{div} 2, j \text{div} 2)$
NW brother	$(k, i-1, j-1)$
N brother	$(k, i-1, j)$
NE brother	$(k, i-1, j+1)$
W brother	$(k, i, j-1)$
E brother	$(k, i, j+1)$
SW brother	$(k, i+1, j-1)$
S brother	$(k, i+1, j)$
SE brother	$(k, i+1, j+1)$
NW sun	$(k+1, 2i, 2j)$
NE sun	$(k+1, 2i, 2j+1)$
SW sun	$(k+1, 2i+1, 2j)$
SE sun	$(k+1, 2i+1, 2j+1)$

In Fig.4 the interconnection schemata between the PE and its SN are shown. In the case of four connections, the set of brothers is composed of $\{N, W, S, E\}$.

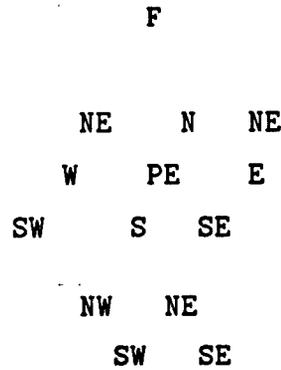


Fig.4 Interconnection schemata.

One limit of this architecture is that the links between the PEs are fixed and the navigation in the structure is expensive and tricky. Further problems can arise because the number of processors are reduced exponentially as soon as the computation goes bottom-up.

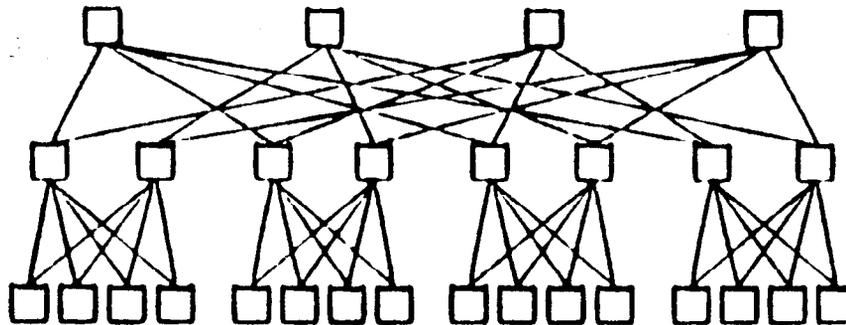


Fig.5 "Interleaved Pyramid" architecture.

The "interleaved pyramid" [19] has been proposed in order to overcome some of these problems. Each non-root node has two (or more) parents (see Fig.5) and the realized network may be considered as a set of interleaved "basic pyramids".

The links are still fixed, but the number of pathways from each descendant to its root is now exponentially increased. If the linear size of the pyramid is 2^L than the number of paths are 2^{L-1} instead of one. The number of processors available at the root for computation is also increased 2^L rather than one.

Some remarks must be made about the implementation of the interleaved pyramid. The first regards the increased hardware complexity, in terms of number of PEs.

The number of PEs required is $2^{2L+1} - 2^L$ instead of the $4/3 \times 4^L$ required for the basic pyramid. For $L > 5$ 50% more PEs are required. The second remark regards the number of interconnections. The number of vertical connections required for the interleaved pyramid is $4^{L+1} - 2^{L+2}$ while for the basic pyramid the number is $4/3 \times 4^L$. This means that for $L > 5$, the number of vertical connections increases more than 150%. A last remark concerns the software and the hardware of the control unit. There is an explosion of the paths needed for more sophisticated concurrency strategy; for example two fathers could contend with the same son.

The "pyramid tree" [20] has been introduced in order to reduce the hardware explosion and to still maintain the same flexibility. The "pyramid tree" is a tree, nodes of which are interleaved pyramids. Therefore for $L = P \times K$, the tree has P levels, each non-leaf node has K levels, and each leaf node has $K - 1$ levels. The complexity study of this pyramid is quite difficult and it depends on many parameters (K,P,A), Here A is the number of edges in the tree. Fig.6 gives a sketch of a "pyramid tree".

Although the divide and conquer strategy reduces the hardware complexity, the control of such a pyramid seems more difficult. As a matter of fact each node of the tree is an interleaved pyramid which may require its own data and instructions set. Moreover an efficient system is needed to control the exchange of data between the interleaved pyramids.

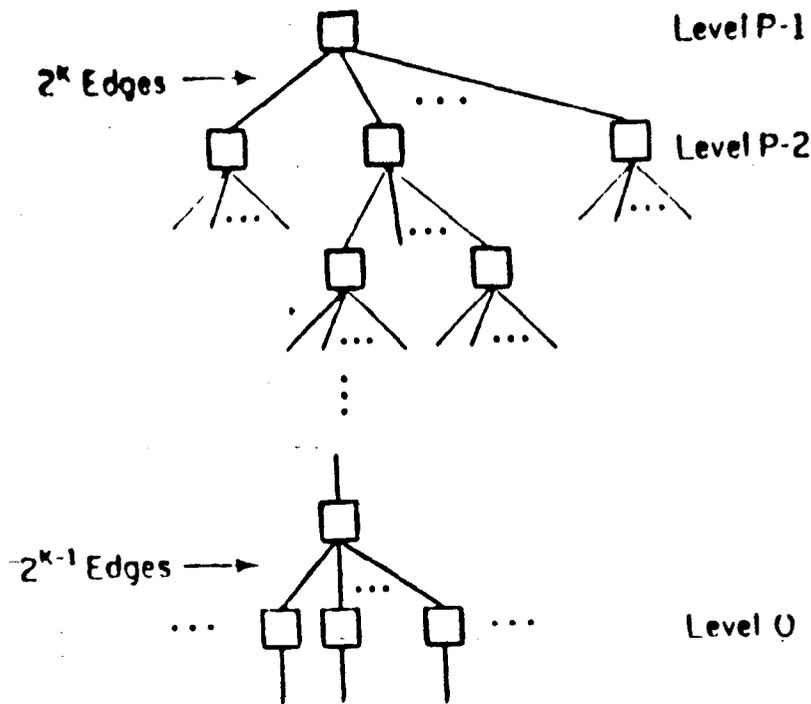


Fig.6 "Pyramid Tree" architecture.

II.2 HARDWARE STRUCTURE

The performance of a pyramid machine depends mainly on the control unit design and on the computing power of the PEs in the array. The hardware complexity of the PEs is related to the complexity of the image functions. If point and local functions are required a very simple and fast PE, performing boolean and serial arithmetic, is enough. An example of a typical low level PE is described in Table 2. If complex high level vision functions are required, more complex PEs are also required containing more internal intelligence (program counter, more bits registers, etc.).

— In the case of elementary PEs a large degree of integration is usually required. As a matter of fact each of them corresponds to a single pixel of the image. In the case of powerful PEs a large region of the image can be relegated to a single processor.

Table 2

technology CMOS or NMOS
two variable length shift registers
three boolean registers
boolean ALU
mask register
RAM (internal memory)
OR-Sum-tree register
I/Os channel to neighbors 9/13

Pyramid machines are classified as homogeneous or etherogeneous depending on whether the PEs are the same or not. In etherogeneous, pyramids each level has specialized PEs to perform a given computation. More powerful PEs will be required at the higher levels of the pyramid; the apex could in fact be a main frame (LISP-machine, VAX, SUN).

II.3 OPERATION MODE

In the operation mode two paradigms of the parallel computation may be considered. In the Single Instruction Multi Data (SIMD) mode all the PEs execute the same instructions on their own data. In Multi SIMD mode (MSIMD) different layers of the pyramid operate in SIMD mode, but with different instruction sets. If the pyramid works in MSIMD mode, pipeline computation may be realized, each layer of the pyramid sending current partial results to the higher layer and receiving data from the lower layer. The results are available at the apex of the pyramid.

Homogeneous pyramids may operate in both SIMD or MSIMD mode, while etherogeneous pyramids may operate only in MSIMD mode.

III. OVERVIEW

This section provides an overview of the relevant projects that are now in progress. For each project a short biography (birth place, birth date, technology, link strategy, operation mode,...) is given.

PCLIP

PCLIP stands for Pyramid Cellular array for Image Processing. The project has been developed at the University of Washington (Seattle, USA). It started in 1982 under S. Tanimoto's leadership [21].

The PCLIP's hardware is homogeneous; each PE has three one-bit registers, one mask register, I/O channels and ports for 13 neighbors. Each PE has allocated 8Kbit of external memory. The technology used is NMOS. PCLIP is a "basic pyramid", the first prototype has 6-levels (base size 32×32). In the first release will operate in SIMD mode. Parallel PASCAL [22] and an assembler will be available to program the PCLIP.

Presently, a single chip containing a PE with all its neighbors has been realized. The control unit and the active memory are still in the progress phase. A simulation of the pyramid has been realized on a Symbolic lisp machine in order to design pyramidal algorithms and to do performance evaluation.

HCL

HCL stays for Hierarchical Cellular Logic; this project has been developed in co-operation between the University of Washington and the National Bureau of Standards. It started in 1984. The two leading scientists are E.W.Kent and S.Tanimoto [23,24]. Hierarchical cellular logic is a set of algebraic and morphological operators applied to objects called bit-pyramids [23]. HCL is a pyramid machine based on this algebra and has been emulated on PIPE^{TM*} (Pipeline Image Processing Engine).

— PIPETM [25] is a set of modular configurable processing stages (MPS), which are connected by concurrent and interacting image-flow pipelines. Results of independent operations on images at each stage are output over for-

* PIPE is a registered trademark of Digital Analog Design Associated, Inc.

ward, recursive and retrograde pathways. The input to each stage may be obtained from any algebraic or boolean combination of the images arriving on the three pathways (see Fig.7).

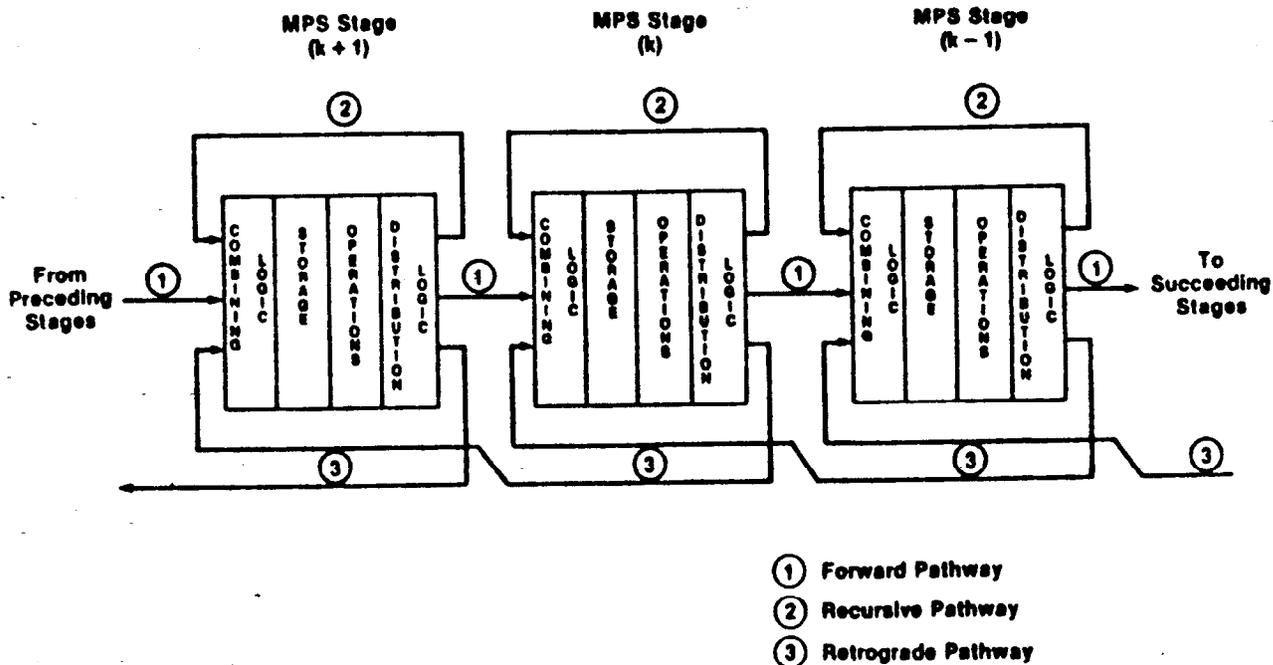


Fig.7 The PIPE processor architecture.

HCL is a "basic pyramid" and each PE has 13 neighbors. Each stage of PIPE may process 256x256x8 images. Therefore the pyramid configuration has 9 levels. HCL operates in both SIMD and MSIMD mode.

MPP PYRAMID (PY-MPP)

This project has been developed at the George Mason University, Virginia, USA. It started in 1984 under the direction of D.H. Schaefer [26]

The hardware is based on the Massive Parallel Processor machine (MPP), which has been designed and realized at NASA/GSFC [27]. MPP is a cellular machine dedicated to IP. The chip has been realized in CMOS technology. In the plane the PEs have four connections. Each PE includes 6 one-bit registers and two 32-bit shift registers for serial arithmetics. Each PE also has one ORSUM and one mask register, 1Kbits of internal memory, and external memory to share data between PEs.

The PY-MPP is a "basic pyramid", that is realized by layering a set of MPP planes. The vertical connections are provided by a set of multiplexing stages. Fig.8 shows the realized architecture. As of now the machine has five levels (base size 16x16).

Users may program the machine in the PLANB language and the assembler languages. The operational mode is SIMD.

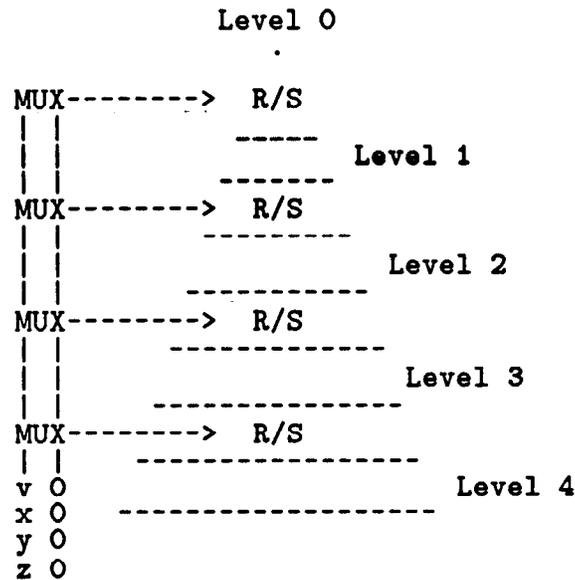


Fig.8 The PY-MPP architecture.

PAPIA

The PAPIA project has been developed as a coordinated program between the Italian Public Instruction Department and the National Research Council. The project started in 1984, and is being developed at the Universities of Milano, Pavia, Palermo, Rome, and at the Institute IFCAI/CNR in Palermo. The leader of the project is S. Levialdi (Univ. of Rome) [28,29].

PAPIA stands for Pyramidal Architecture for Parallel Image Analysis and is a "basic pyramid" with four connections in the plane. Therefore each PE has 9 neighbors. The chip technology is NMOS. Each PE has two one-bit registers, two 16-bits shift registers for serial arithmetic, one ORSUM, and one mask register. Each PE also has 256 bits of internal memory, and I/O channels to the neighbors. PAPIA will operate in both SIMD and MSIMD modes. In its first

release it will have 6 levels (base size 32x32) and will operate in SIMD mode. For the present, those parts concerning the chip and controller hardware have been completed. The software environment is in progress. For now users may write PAPIA programs in assembler and PMACRO languages. The latter one allows one to insert macro assembler instructions in a main program written in C-language. A Pyramid C Language (PCL) is under development [30].

SPHINX

SPHINX stands for Systeme Pyramidal Hierarchise pour le Traitement d'Image Numeriques. The project was started at the Paris-Sud University, France, in 1985 [31]. The PEs are homogeneous and have four connects in the plane. The verticals links are arranged in binary tree fashion. It follows then, that each PE has one father and two sons for a total of seven neighbors. This connection scheme reduces the total number of connections required in the chip.

An experimental version of the PE has been integrated in NMOS; a CMOS version is currently being planned. Each PE has 3 one-bit registers, one mask register, and 64bits of internal memory. No information is yet available about the number of levels. SPHINX operate in SIMD mode and has only assembler language for program development. At present both chip and control unit hardware are in the developmental stage. The pyramid machines so far presented are homogeneous and are classified as "basic pyramids". We end this section with a short discussion of two more hierarchical machines whose architecture is quite different.

FLAT PYRAMID

This project has been proposed at the University of Wisconsin, USA, in 1985 under the direction of P.A. Sardon [32]. The FLAT pyramid has been introduced in order to mitigate the bottleneck present at the top level of homogeneous pyramids. The aim of the project is to built a pyramid with PEs, for which the power increases with the level. It is composed of a single reconfigurable array of $N \times N$ 1-bit PEs, $N/2 \times N/2$ 4-bits PEs, ..., 1×1 $N \times N$ -bits PE. Each configuration corresponds to one layer of the pyramid. Each

PE has several 1-bit registers and communication buses. An ALU provides for logic and arithmetic; the memory is external.

The hardware complexity is increased by two factors: The fact that neighborhoods change as the configuration is changed and that the memory associated with each PE must be also reconfigurable. A single controller issues instructions to all PEs in the reconfigurable array and it is responsible for the reconfiguration of the array.

The potential advantages of the FLAT pyramid may be summarized as follows: 1) the reconfiguration of the PEs allows more powerful processors to be used in the upper layers of the pyramid; 2) the vertical connections, between pyramid layers are simplified; 3) the number of processors required is less than those required by a conventional pyramid.

Disadvantages are: 1) the parallelism between layers is lost; 2) only one pyramid layer at the time is active; 3) the horizontal connections are complex; 4) the hardware implementation is very difficult, given current VLSI technology.

NETRA

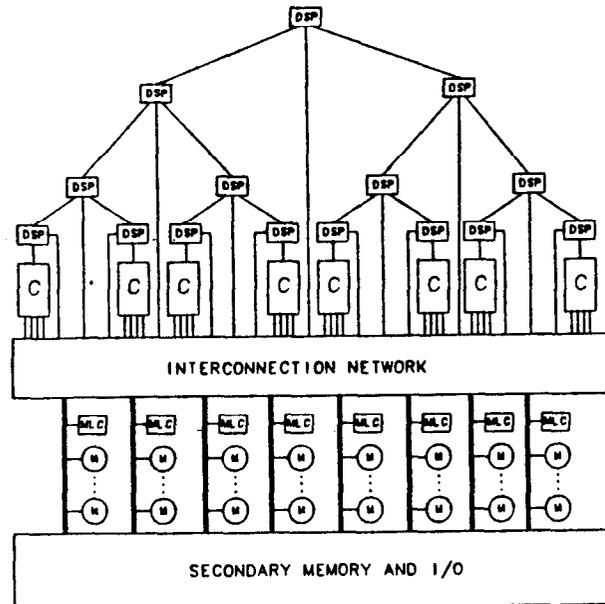
NETRA is a multiprocessor architecture dedicated to handling low and high level computer vision problems. It has been proposed at the University of the Illinois, USA, in 1984 [33]. NETRA consists of the following components:

- a large number of PEs ($10^2 - 10^4$) organized in clusters of 16 to 64 elements each;
- a tree of Distributing and Scheduling Processors (DSP) that distribute the control to the processors;
- a parallel pipelined Global Memory;
- an interconnection network that links the PEs and DSPs to the global memory.

The system is illustrated in Fig.9.

The hierarchical organization of the DSPs allows the realization of a data driven computing. Each cluster of PEs can be specialized for a particular

function and it can cooperate with other clusters under the control of a hierarchy of controllers. Complex strategies are allowed by the pyramid set of NETRA control units associated with the clusters of powerful PEs.



C Cluster
M Memory
DSP Distributing and Scheduling Processor
MLC Memory Line Controller

Fig.9. Organization of NETRA.

IV.COMPARISON NOTE

In the previous sections we have shown some of the main projects for building machines with pyramidal and hierarchical architecture.

Some of them are at a proposal stage (FLAT PYRAMID and NETRA), while others are at an implementation phase (PCLIP, PAPIA, SPHINX); very few are working (PY-MPP and HCL). However we emphasize that PY-MPP is at a very preliminary prototype stage and HCL is a simulated version of a basic pyramid on an existing parallel machine (PIPE).

Therefore a full comparison between all the pyramid machines, based on:

- hardware performance and modularity;

- neighbors configuration;
- links flexibility;
- software environment;
- algorithms benchmark.

seems to be premature.

Table 3 gives a rough idea of the main features and differences existing between the pyramid projects that have been discussed. FLAT PYRAMID and NETRA have not been included because both are at the proposal stage and have a quite different architecture. HCL has not been considered because it is based on an existing hardware and can be considered more as a hardware simulation of a pyramid machine.

Table 3

	PY-MPP	PCLIP	SPHINX	PAPIA
TECHNOLOGY	CMOS	NMOS	NMOS	NMOS
# PINS/CHIP	52	64	44	48
1bit REG	6	3	3	2
# AND LENGTH	32bits	No	No	16bits
SHIFT REG	2	No	No	2
MASK REG	Yes	Yes	Yes	Yes
OR-SUM-REG	Yes	Yes	No	Yes
INT.MEMORY	1Kbits	No	67bits	256bits
EXT.MEMORY	Yes	No	No	Yes
# NEIGHBORS	9	13	7	9
OP.MODE	SIMD	SIMD	SIMD	SIMD
LANGUAGES	PLANB	PPASCAL	No	PCL

Some remarks concerning Table 3 are appropriate. PCLIP seems to have the best neighbor connection scheme, allowing easier and faster implementations of algorithms requiring mainly local computation. On the other hand, PCLIP has no shift registers, therefore the implementation of arithmetic is

harder. SPHINX has the merit of simple interconnection schemata, however it has no shift and OR-SUM-TREE registers, the lack of which creates big problems in the control of the flux of parallel algorithms. PY-MPP has a good arithmetic, big internal memory. Some restrictions derive from multiplexer channels for the vertical navigations of the data. In fact the handling of the multiplexors is external to the array and very slow. PAPIA has a good serial arithmetic, but it is less powerful than PY-MPP.

V. CONCLUDING REMARKS

Theoretical considerations and experimental simulation studies have demonstrated that some increase in speed can be derived from the use of pyramids in Image processing and analysis. The implementation of such parallel 3D machines has become feasible owing to the advent of modern VLSI technology. However, up to now only low level vision problems seem to profit from the use of pyramid machines. In order to realize fully the integration between low and high level vision some requirements must be satisfied. There is the need for new development in VLSI technology in order to realize more flexible and wider linking strategies. The computational capability of the PEs must increase with their functional complexity. This means that one must design and build not-homogeneous systems. Hierarchical and functional systems, which include data-driven computation (see NETRA), seem to allow a natural and efficient implementation of systems for the analysis of high level vision problems.

Much effort must be made to develop high level languages. These languages must include appropriate data structures (type image, pyramid, mask,...) and instructions sets that allow the easy implementation of all the local operations and the navigation (horizontal and vertical) in the pyramid.

ACKNOWLEDGEMENTS

The author gratefully thanks Mrs. Arla Le Count for her help in the editing of the paper, and J.H. Friedman for the improvement of the english.

REFERENCES

- [1] K. Preston Jr., "The CELLSCAN system - a leucocyte pattern analyzer", in Proc. Western Joint Computer Conf., p.173, 1961.
- [2] H.J. Siegel, "PASM: a Partitionable SIMD/MIMD system for Image Processing and Pattern Recognition", IEEE Trans. on Computer, Vol C-30, pp.934-947, 1981.
- [3] S.F. Roddaway, "DAP-A distributed processor array", First Annual Symp. on Comp. Architect., Florida, 1973.
- [4] M.J. Duff, "Review of the CLIP image processing systems", Proc. Nato Conf. Anaheim, USA, 1981.
- [5] V. Cantoni, S. Levialdi, "Matching the task to an image processing architecture", Graphics and Image Processing, Vol.22, pp.301-309, 1983.
- [6] V. Cantoni, "Classification schemes for image processing architectures", NATO ASI on Computer Architectures for Spatially Distributed Data, Cetraro, 1983.
- [7] S.L. Tanimoto, "Regular Hierarchical Image and Processing Structure in Machine Vision", Comp. Vision Systems, ed. A.R. Hanson and E.M. Riseman, Academic Press N.Y., pp.165-174, 1978.
- [8] L. Uhr, "Recognition Cones and Some Test Results", Comp. Vision System, ed. A.R. Hanson and E.M. Riseman, Academic Press, N.Y., pp.363-372, 1978.
- [9] R.A. Finkel, J.L. Bentley, "Quadrees: a data structure for retrieval on composite keys", Acta Informatica, 4, 1974.
- [10] J.H. Friedman, J.L. Bentley, R.I. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time", STAN-CS-75-482, Stanford University, 1975.
- [11] A. Rosenfeld, "Quadtree and pyramids for pattern recognition and image processing", 5th Intl. Conf. on Pattern Recognition, Miami Beach, 1980.
- [12] H. Sammet, "Region representation quadtree from boundary codes", Communication ACM, Vol.23, 1980.

- [13] C.R. Dyer, "The space efficiency of the quadtrees", *Computer Graphics and Image Processing*, Vol.19, pp.335-348, 1982.
- [14] L. Uhr, "Layered recognition cone networks that preprocess, classify and describe", *IEEE Trans. Comput.*, C-21, 1972.
- [15] S.L. Tanimoto, T.A. Pavlidis, "A hierarchical data structure for picture processing", *Comp. Graphics & Image Processing*, Vol.4, 1975.
- [16] S.L. Tanimoto, "Sorting, histogramming, and other statistical operations on pyramid machines", *Multiresolution Image Processing and Analysis*, A. Rosenfeld ed., Springer Verlag, NY, 1983.
- [17] V. Di Gesu', A. Machi', "Some examples of IP algorithms for pyramidal architecture", *III Intl. Conf. on Image Analysis and Processing*, Rapallo, 1985.
- [18] L. Uhr, "Psychological motivation and underlying concepts", in S.L. Tanimoto and A. Klinger (Eds.), *Structured Computer Vision: Machine Perception through Hierarchical Computation Structure*, Academic Press, NY, pp.1-30, 1980.
- [19] N. Ahuja, S. Swamy, "Interleaved pyramid architectures for bottom-up analysis", *Proc. of the 6th Intl. Conf. on Pattern Recognition*, pp.388-390, Munich, 1982.
- [20] N. Ahuja, S. Swamy, "Multiprocessor pyramid architectures for bottom-up analysis", *IEEE Trans. on PAMI*, Vol.6, No 4, pp.463-475, 1985.
- [21] S.L. Tanimoto, "A pyramidal approach to parallel processing", *Proc. 10th Annual Intl. Symposium on Comp. Architecture*, Stockholm, Sweden, pp.372-378, 1983.
- [22] A.P. Reeves, "Parallel PASCAL: an extended PASCAL for parallel computers", *Jnl. of Parallel and Distributed Computing*, Vol.1, pp.64-80, 1984.
- [23] S.L. Tanimoto, "A hierarchical cellular logic for pyramid computers" *Jnl. of Parallel and Distributed Computing*, Vol.1, pp.105-132, 1984.

- [24] E.W.Kent, S.L.Tanimoto, "Hierarchical Logic and PIPE processor:Structural and Functional Correspondence", Proc. IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management", pp.311-319, Maiemi, 1985.
- [25] E.W. Kent, M.O. Shneier, R. Lumia, "PIPE: Pipeline Image Processing Engine", Jnl. of Parallel and Distributed Computing, Vol.2, pp.50-78, 1985.
- [26] D.H. Schaefer, D.H. Wilcox, G.C. Harris, "A pyramid of MPP processing elements - experiences and plans", Proc. 18th Annual Hawaii Intl. Conf. on System Sciences, Vol.1, pp.178-184, 1985.
- [27] K.E. Batcher, "Design of a Massively Parallel Processor", IEEE Trans. Computers, Vol. C-29, pp.836-840, 1980.
- [28] S. Levialdi, "The PAPIA project", 2nd Intl. Workshop on Data Analysis in Astronomy, Erice, 1986.
- [29] V. Cantoni, L. Carrioli, O. Catalano, L. Cinque, V. Di Gesu', L. Ferretti, G. Gerardi, S. Levialdi, R. Lombardi, A. Machi', R. Stefanelli, "The PAPIA image analysis system", SPIE Intl. Symp. on Image Processing, Cannes, 1985.
- [30] V. Di Gesu', "An high level language for pyramidal architectures", NATO ARW on Pyramidal Systems for Image Analysis and Computer Vision, Maratea, 1986.
- [31] A. Merigot, B. Zavidovique, F. Devos, "SPHINX, a pyramidal approach to parallel image processing", Proc. IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management", pp.107-111, Maiemi, 1985.
- [32] P.A. Sandon, "A pyramid implementation using a reconfigurable array of processors", Proc. IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management", pp.112-118, Maiemi, 1985.
- [33] M. Sharma, J.H. Patel, N. Ahuja, "NETRA: an architecture for a large

scale multiprocessor vision system", Proc. IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management", pp.311-319, Maiemi, 1985.