Highly Parallel Information Processing System for Graphical Data Processing and Associative Processing

AKIO TOJO* AND TETSURO YAMAGUCHI*

1. *Introduction*

Many of the methods studied in the fields of character recognition or pictorial pattern recognition aim at realizing machines which can determine to which class out of a finite set of classes a given pattern belongs, and the answer obtained with such a machine is either the name of the class the pattern belongs to or the response that indicates the machine is unable to classify the object.

It is almost impossible, however, to perform processing of graphic data other than characters only with such approaches. This will be clear if one thinks of problems of processing graphic data obtained from the natural world, such as photographs of particle tracks in a bubble chamber or spark chamber, finger prints, photographs of chromsomes or neurons, or air photographs, and problems of processing complicated general graphic patterns including circuit diagrams, flow charts or chemical formulae, etc. [1]~[4].

The system required with such problems is the one which can extract common features like cross points, end points, line segments, or shade and tone from the objective pattern, and convert it into a structured data or a description feasible to processing based on the topological or positional relationships and some other attributes of them, and the one which is able to store, reproduce and recognize the object when requested. Digital computers presently in use, however, do not have well suited functions required for such operations. One of the objectives developing the present system described here is to endow the general purpose computer with an ability to process these kind of problems quite easily by adding the developed system.

2. *Background of the problem*

There have been several attempts to apply language processing techniques to the problems of pattern recognition or picture processing [5], [6]. Kirsh and Narasimhan pioneered in this field [7], [8]. Recently formal treatment of pictorial patterns using picture description language and its application to picture recognition systems or picture generation systems are reported [9], One of the

This paper first appeared in Japanese in Joho-Shori (the Journal of the Information Processing Society of Japan), Vol. 11, No. 2 (1970), pp. 70–76.

* Information Systems Section, Electrotechnical Laboratory, MITI, Tokyo.

authors also proposed similar methods and basic system design for pictorial pattern processing [10]. This paper gives a general description of the basic construction and functions of the experimental system and some of the results of simulation of the system.

In the field of pictorial pattern processing and computer graphics, one of the most important problems is how to express and manipulate picture elements and their relationships. The system in which logical structure of an information and physical locations where its elements are stored are specified seperately in order to facilitate handling relations between data elements is called associative processing system. In the most of the interactive graphic systems, data structures with this property are employed [11].

Associative processing system will be realized quite easily if a content addressable memory (CAM) in available, but at present number of problems have to be solved before this become practical.

One approach to realize CAM is to simulate content addressability using programming techniques such as hash coding. Several works have been reported in this area [12], [17]. Hardware CAMs are also tried strenuously utilizing LSI technology and others [13]. Another has proposed to modify 2 1/2D magnetic core storage to obtain content addressability [14].

The present system described here also possesses an ability to detect data with a specified bit pattern in parallel regarding $n \times n$ bits two dimensional data in the processor as n word n bits information, and this will be usefull in realizing associative processing system for pictorial pattern processing [15], [16].

There have been proposed various types of highly parallel systems for varying purposes [18]. ILLIAC III is the most famous and has been the only project attempting to implement hardware system [19]. Although the system described here is experimental and rather small in scale, it has a number of new features in its basic functions or system construction, and it will be possible to reinforce the ability of the present computer system by including this type of processor in it. [15], [16], [20].

3. *System construction*

The central part of the system is the processor with parallel structure (AIPU —Associative Information Processing Unit) in which identical logic units (logic cells) with relatively simple logical functions are arranged into $n \times n$ array so that information can be exchanged between the nearest neighbor cells. To these logic cells common control signals are fed from the AIPU control.

Each logic cell consists of 4 bit register and associated logics, and each bit position of the register can be considered to compose a planar register holding a two dimensional information. They are called PLANE_0, PLANE_1, PLANE_2,

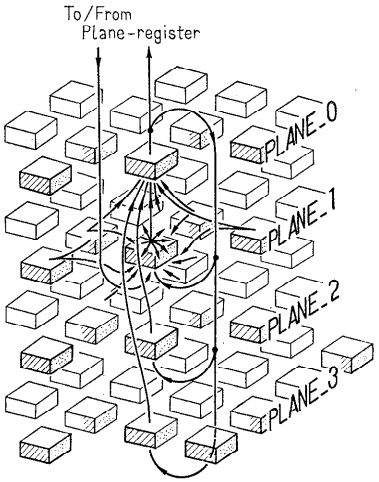


Fig. 1. Basic construction of Associative Information Processing Unit. Arrows show main information flow in a cell. Flow of control signals is not included.

and PLANE_3. Fig. 1 shows the flow of informations in a cell in this processor.

The PLANE-REGISTER is a storage device with a very large word length and $n \times n$ bit data can be transferred in parallel between AIPU and this device. Through the periphery of a plane in the processor, n bit data can be transferred to and from the main memory of the computer to which AIPU is attached.

AIPU control which is also attached to the computer to decode instructions supplied and generate control signals according to the instructions possesses two basic registers, R_REGISTER and D_REGISTER. R_REGISTER is composed of a pair of eight bit cyclic shift registers and D_REGISTER a pair of n bit registers. Both are used to hold a reference patternfor two dimensional local pattern matching operation (template matching operation) or for one dimensional matched search operation respectively. Outputs from R_REGISTER are applied to all of the cells in AIPU in common, while outputs from each digit of D_REGISTER are only applied to the cells in the corresponding column of AIPU.

Total of 35 control signals including the ones from above mentioned registers are fed to each logic cell [16], [20]. Input data for each cell are eight from eight neighboring cells and one from PLANE_REGISTER. Output from a cell is obtained from PLANE_0 or PLANE_1.

In the system now under development, n is 16 and each cell is composed of 15 commercially available current switching integrated logic circuits contained in one printed circuit board.

Fig. 2 shows the principle of control signal distribution through AIPU. The time deviation of control signals on the plane are restricted within 4 to 5 ns.

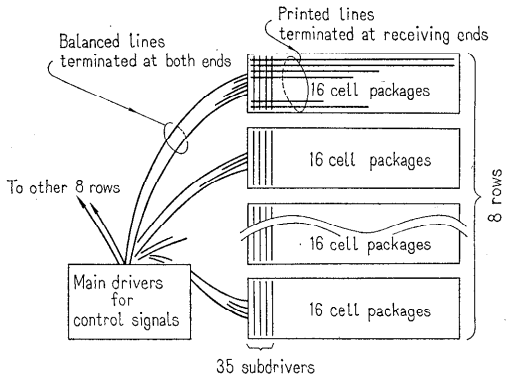


Fig. 2. Control signal distribution scheme for AIPU.

Lines for interconnections between cells are not terminated.

Employing the packaging scheme mentioned above, it has been shown that the response time of each cell to control signals is less than 10 to 25 ns [20]. As basic operations in AIPU described in the next section consist of 2 to 30 steps of control sequences, the system is expected to operate sufficiently fast compared with the presently used main memory system.

4. *Basic operations in the system*

A set of 19 instructions is prepared for AIPU operation as shown in Table 1.

Table 1. AIPU instruction repertoire and format.

INSTRUCTION	OP CODE (OCTAL)	FORMAT
PLANE_AND	07	OP CODE, CL, D1, D2, D3
PLANE_OR	06	
AND_TEMPLATE	05	OP CODE, STEP, RV, CL, D1, D2, D3
OR_TEMPLATE	04	
BITWISE_AND	03	OP CODE, CL, D2, D3
BITWISE_OR	02	
PROPAGATE	01	OP CODE, D2, D3
SHIFT_LOAD	016	OP CODE, SD, CH, CV, N
STORE_SHIFT	014	
SHIFT	010	OP CODE, CH, CV, N
PARALLEL_LOAD_FROM_PLANE_REGISTER	013	OP CODE, PR ADDRESS
PARALLEL_STORE_TO_PLANE_REGISTER	011	
LOAD	006	OP CODE, H, V, SRC, DEST, SRC, DEST, RR, RR
STORE	004	
PARALLEL_INTERNAL_TRANSFER	007	OP CODE, CLR, DEST
PARALLEL_INTERNAL_TRANSFER_NEGATED	005	
SET_RR_UPPER_HALF	003	OP CODE, RR1/5, RR2/6, RR3/7, RR4/8
SET_RR_LOWER_HALF	001	
NO_OPERATION	000	OP CODE

These instructions are supplied from the general purpose computer to AIPU control through a high speed data channel and executed in AIPU or the associated systems. Some of the fundamental operations are explained below.

(1) *PLANE_AND/OR, BITWISE_AND/OR*

By this operation, logical product or sum of the content of PLANE_1, PLANE_2, and PLANE_3 is obtained in PLANE_0 logically added to the former content of PLANE_0. CL bit specifies to clear PLANE_0 prior to this operation, and *D* bits specify negation or 'don't care' condition.

For example, PLANE_AND operation with $D1$='10', $D2$='11', and $D3$='01' gives the following

$$[\text{PLANE\_0}] \leftarrow \overline{\text{CL}} \cdot [\text{PLANE\_0}] + [\text{PLANE\_1}] \cdot [\overline{\text{PLANE\_3}}]$$

where [ ] represents the content of each plane.

As will be seen from this example, '11' indicates 'don't care' condition and the content of the plane corresponding to this condition is completely neglected.

Bitwise Boolean operation is similar to this operation except that negation or 'don't care' condition for PLANE_1 is specified bitwise by the content of D_REGISTER.

(2) *AND/OR_TEMPLATE*

This operation is for two dimensional local pattern matching to extract points in PLANE_1 which compose local patterns with their eight nearest neighbor points that match to a given reference pattern (3×3 template matrix) as shown in Fig. 3.

$r_{-1,-1}^{(*)}$	$r_{-1,0}^{(*)}$	$r_{-1,1}^{(*)}$
$r_{0,-1}^{(*)}$	$d_1^{(*)}$	$r_{0,1}^{(*)}$
$r_{1,-1}^{(*)}$	$r_{1,0}^{(*)}$	$r_{1,1}^{(*)}$

Fig. 3. 3×3 template matrix. Asterisks denote 0 or 1.

The elements of the template matrix except the one in the center are held in R_REGISTER. The center element of the tempelate is given by $D1$, $D2$ and $D3$ bits in the instruction. All of these elements are composed of 2 bits each and can specify '1', '0' or 'don't care' conditions.

During the execution of template matching operation, the template is rotated with a step specified by the STEP bits of the instruction, and the result is a logical sum of the matched conditions obtained in PLANE_0. In the case of AND_TEMPLATE instruction, for example, the following operation will be executed at each step for all combinations of (i, j) and (k, l) pair

$$p_{0,i,j} \leftarrow p_{0,i,j} + \underset{k,l}{\pi}(r_{k,l}{}^{(0)} \cdot \overline{p_{1,i+k,j+l}} + r_{k,l}{}^{(1)} \cdot p_{1,i+k,j+l}) \underset{t}{\pi}(d_t{}^{(0)} \cdot \overline{p_{t,i,j}} + d_t{}^{(1)} \cdot p_{t,i,j})$$

$$\begin{pmatrix} k,\ l=0,\ \pm 1 \text{ (except } k=l=0). \\ t=1,\ 2,\ 3. \quad i,\ j=1,\ 2,\ 3,\ \cdots,\ n \end{pmatrix}$$

where $p_{t,i,j}$ denotes the status of a cell on PLANE_t at coodinates (i, j), and π represents logical product. $r_{k,l}{}^{(0)}$ and $r_{k,l}{}^{(1)}$ represent the content of R_REGISTER, and $d_t{}^{(0)}$ and $d_t{}^{(1)}$ are given in the Dt bits of the instruction as mentioned above. If RV bit in the instruction is true. these operations will be repeated for the mirror symmetric template.

In the case of OR_TEMPLATE, the operation at each step will be expressed by the following equation

$$p_{0,i,j} \leftarrow p_{0,i,j} + \sum_{k,l}(\overline{r_{k,l}{}^{(0)}} + \overline{p_{1,i+k,j+l}}) \cdot (\overline{r_{k,l}{}^{(1)}} + p_{1,i+k,j+l})$$

$$+ \sum_{t}(\overline{d_t{}^{(0)}} + \overline{p_{t,i,j}}) \cdot (\overline{d_t{}^{(1)}} + p_{t,i,j})$$

where $\sum$ represents logical sum.

In these operations, results may not be defined on the periphery of the plane. Four $n+1$ bit registers to hold boundary bit patterns are prepared around PLANE_1 to overcome this situation. Actually two of these boundary registers serve also as D_REGISTER.

(3) *PROPAGATE*

With this operation, the statuses of '1' cells on PLANE_1 are propagated by rippling to the periphery of the plane in the direction specified by the content of R_REGISTER. The region in which the propagation occurs can be restricted by the content of PLANE_2 and PLANE_3 setting $D2$ and $D3$ bits in the instruction appropriately.

For example, in case $D2=D3=$'11', propagation can occur on all over PLANE_1, while if $D2=$'10' and $D3=$'01', it can occur only on the region identical to the pattern given by [PLANE_2]·$\overline{[\text{PLANE\_3}]}$.

The propagation operation can be used in checking the status of planes, extracting parts of a given pattern, obtaining a projection of a pattern, analyzing positional relationships among pattern elements, or matched search of one dimensional data.

(4) *Other operations*

SHIFT instruction is prepared to move a pattern on a plane in the direction specified by the content of R_REGISTER. SHIFT_LOAD and STORE_SHIFT are used to transfer data between AIPU and the main memory in the computer.

The content or negated content of PLANE_0 can be transferred to PLANE_1, PLANE_2, or PLANE_3 by PARALLEL_INTERNAL_TRANSFER instruction.

Contents of boundary registers and D_REGISTER can be set by LOAD instruction, and the content of R_REGISTER by SET_RR instruction.

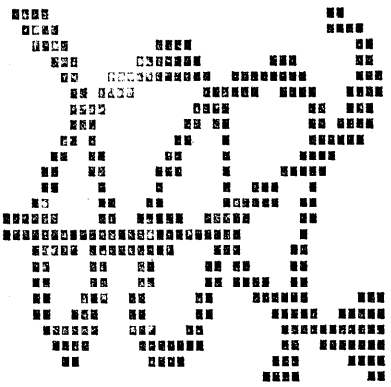
5. *Some of the fundamental operations realized using basic operations*

There are several important operations for two dimensional data processing and associative processing that can be realized utilizing basic operations in AIPU. These are

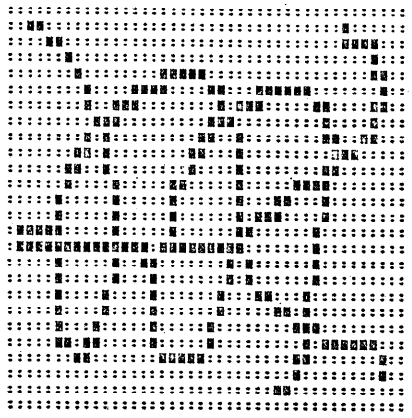
(1) temprate matching operation using a template matrix with a larger dimension than 3×3,
(2) to process a two dimensional data larger than $n\times n$ bits,
(3) one dimensional matched data search,
(4) to change the status of the cell at specified location,
(5) to check the status of a plane.

For the details of algorithms to realize these operations refer to references [16] and [21].

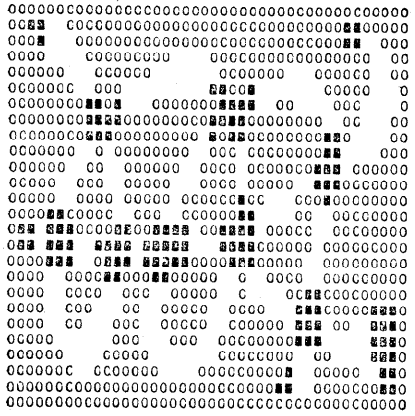
6. *Simulation of the system*



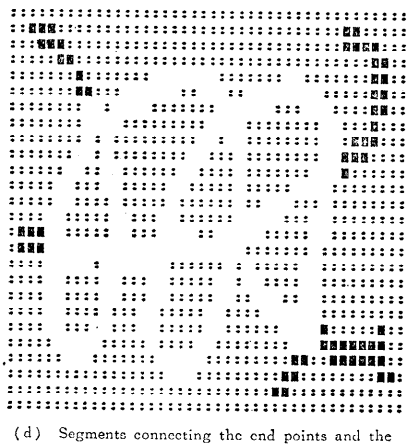
(a) Original pattern



(b) Skeleton obtained with the most simplified algorithm



(c) Cross points and end points detected



(d) Segments connecting the end points and the nearest cross points including the end points

Fig. 4. Results of excuting some of the fundamental operations required for binary pattern analysis using AIPU simulation program.

In realizing the system described above, experiments on basic operations required for two dimensional binary pattern analysis have been carried out using AIPU simulation program. Some of the results are shown below.

Fig. 4(b) is an example of extracting a skeleton from the pattern shown in Fig. 4(a).

Fig. 4(c) shows an example of detecting cross points and end points in the original pattern. It is also possible to detect the cross points and the end points separately. These operations are realized employing 3×3 template matching operation and parallel boolean operation.

Fig. 4(d) presents the result of extracting segments linking these end points and the nearest cross points. This operation is realized by propagation.

(a) Original pattern (b) Result

Fig. 5. Detection of abrupt change in thickness.

Fig. 5(b) shows the result of locating points in the given pattern where thickness abruptly changes.

All of these operations, if possible to be carried out without dividing the original pattern, can be realized with several to several tens of steps of basic instructions in AIPU. The details will be found in references [10] and [16].

7. *Conclusion*

The highly parallel processor now under development consists of AIPU with 16×16 cells, AIPU control, and a long word plated wire memory employed for PLANE_REGISTER. The experimental system is connected to NEAC 3200 (16 bits, 16 kwords, 960 ns cycle time) through high speed data channel, and operates as a sort of harwared subroutines.

As mentioned before, the unit cell in AIPU is composed of fifteen integrated circuits. It will be feasible, however, to reduce this number to three or four by employing custom-made integrated circuits which utilize techniques of com-

mercially available medium scale integration.

The major fields of application of this system are analysis and conversion to descriptions of binary pictures, and parallel search of one dimensional data. Besides these, it has been shown that the system can be utilized in some kind of pattern synthesis problem such as the one to find a connecting path between two points on a plane with obstacles [22]. The parapropagation character recognition can also be realized with this system [23].

Acknowledgement

The authors are greatly indebted to Dr. K. Noda, Chief of Electronic Computer Division, for his constant support to this project. They are also indebted to Dr. H. Nishino, Chief of Software Division, Dr. O. Ishii, Chief of Memory Systems Section, and Mr. S. Watanabe, Chief of Man-Machine System Section, for their stimulating discussions and helpful comments. The cooperations by Mr. H. Aoyama and Mr. Y. Akiba in implementing the system are also gratefully acknowledged.

References

[1] Wadsworth, B. F., PEPR—A Hardware Description, *Proc. Emerging Concepts in Computer Graphics,* 41-70, 1967.

[2] Wegstein, J. H., A Computer Oriented Single-Fingerprint Identification System, *NBS Technical Note* 443, (March 1968).

[3] Ledley, R. S., L. S. Rotolo, et al., Pattern Recognition Studies in the Biomedical Sciences, *Proc. SJCC,* 411-430 (1966).

[4] Hawkins, J. K., G. T. Elerding, et al., Automatic Shape Detection for Programmed Terrain Classification, *Proc. Filmed Data and Computer Seminar,,* XVI-1-XVI-9 (June 1966).

[5] Feder, J., The Linguistic Approach to Pattern Analysis—A Literature Survey, *Technical Repoort* 400-133, New York Univ., (Feb. 1966).

[6] Miller, W. F. and A. C. Shaw, Linguistic Methods in Picture Processing—A Survey, *Proc. EJCC,* 279-290 (1968).

[7] Kirsch, R. A. Computer Interpretation of English Text and Picture Patterns, *IEEE Trans.,* EC-13, No. 4, 363-376 (1964).

[8] Narashimhan, R., Labeling Schemata and Syntactic Descriptions of Pictures, *Information and Control,* 7, 151-179 (1964).

[9] Shaw, A. C., The Formal Description and Parsing of Pictures, Technical Report SLAC-84, *Stanford Linear Accelerator Center, Stanford Univ.* (March 1968).

[10] Tojo, A., Pattern Description with Highly Parallel Information Processing System. (I), *Bul. Electrotech. Lab., 31,* No. 8, 930-946 (1967).

[11] Tojo, A., Computer Aided Design, *Electronics (Japanese), 13,* No. 13, 1423-1437 (1968).

[12] Feldman, J. A., Aspects of Associative Processing, *Technical Note* 1965-13, Lincoln Lab., MIT, Apr. 1965.

[13] Narud, J. A. and W. C. Seelbach, High Speed Integrated Circuit Memories, Proc. *Conf. on Integrated Circuits,* 29-38, May 1967.

[14] Stones, H. S., Associative Processing for General Purpose Computers through the Use of Modified Memories, *Proc. FJCC,* 949-955, 1968.

[15] Tojo, A. and T. Yamaguchi, Pattern Description with Highly Parallel Information Processing System. (II)—Construction of a Highly Parallel Processor, 1968 Four Electric Institutes Joint Convention Records, No. 2588.

[16] Tojo, A. and T. Yamaguch, Pattern Description with Highly Parallel Information Processing System. (VI)—Construction and Simulation of the System., Bul. Electrotech. Lab., Vol. 35, No. 5, 479-505, 1969.

[17] Ash, W. L. and E. H. Sibley, TRAMP; An Interpretive Associative Processor with Deductive Capabilities, *Proc. ACM National Conf.*, 143-156, 1968.

[18] Murtha, J. C., Highly Parallel Information Processing Systems, in Advances in Computers, Vol. 7, 1-116, Academic Pr., 1966.

[19] McCormick, B. H., The Illinois Pattern Recognition Computer—ILLIAC Ⅲ, *IEEE Trans., EC-12*, 6, 791-813 (1963).

[20] Yamaguchi, T. and A. Tojo, Cell Construction and Operation for Highly Parallel Processor, (1969) Four Electric Institutes Joint Convention Records, No. 3168.

[21] Tojo, A. and T. Yamaguchi, Pattern Description with Highly Parallel Information Processing System. (IV)—Generalized Discussion on Template Matrix Matching Operation., *Bul. Electrotech. Lab.*, *31*, No, 12, pp. 1294-1301, 1967.

[22] Tojo, A., Pattern Description with Highly Parallel Information Processing System. (V)—Distance Function and Minimum Path Connection, ibid., 32, 9, 930-942 (1968).

[23] Glucksman, H. A., A Parapropagation Pattern Classifier, *IEEE Trans., EC-14*, 3, pp. 434-443, 1965.