

# IMAGE PROCESSING BY EXPERIMENTAL ARRAYED PROCESSOR

Hitoshi Matsushima      Takeshi Ueno      Masakazu Ejiri

Central Research Laboratory  
Hitachi Ltd.,  
Higashi-koigakubo 1-280,  
Kokubunji, Tokyo 185, Japan

An image processor based on an arrayed structure was implemented. It has five design features: (1) memory unit separation into a picture memory for images and a control data memory for programs, (2) processing unit separation into a parallel processing part and a serial processing part, (3) parallel processing by an arrayed structure, (4) use of a special micro-processor for an array's processing element, and (5) direct control of parallel processing by programming. It deals with images consisting of 256 x 256 pixels and processes them by sixteen processing elements. The original images are given by a TV camera. The processed images are outputted to CRT display equipments. The basic operations in the array are data conversions, local operations, sequential processings, manipulations between two images and so on. By using those operations, many kinds of image processing can be done in this processor.

## 1. INTRODUCTION

Many proposals for pattern processing have been made to date, and some have been practically implemented, especially for certain industrial and medical applications, in which special processors such as image analyzers are in practical use. However, most image processors can only deal with binary images.[1]

On the otherhand, recently large-scale pattern processors have been studies in some research laboratories.[2] However there are still problems in developing image processors for multi-leveled images. For example, processors tend to become too big and expensive.

However, recent developments in

-----  
This work was contracted with the Agency of Industrial Science and Technology, Ministry of International Trade and Industry, as a part of the National Research and Development Program "Pattern Information Processing System"

commercially available IC's have had a profound effect on the size, speed, and cost of hardware. In addition, the need for general purpose image processors is getting stronger. Thus, the image processor presented here was experimentally developed. The outstanding feature in our image processor IP is parallel processing ability.

This paper described the principles and basic operation in this IP.

## 2. DESIGN PLINCIPLES

The main points considered in establishing our IP are discussed below.

Information included in images depends on the two-dimensional arrangement of pixel's values. So two-dimensional memory access is employed for data utilization. Therefore, the memory was separated into two distinct parts, one for images and the other for control data. The former has useful functions for parallel processing of images.

In image processing, the same operations are repeated many times. In these cases, increasing processing speed by ordinary computer architectures is not efficient. Therefore, the execution unit in our IP was separated into a conventional mini-computer's central processing unit and a parallel processing part. The former mainly deals with control data and sometimes processes pixels one by one, while the latter usually processes pixels by the parallel processing.

In the parallel processing part, an arrayed structure was adopted, mainly to increase processing speed. In pre-processing of feature-extraction of images the same operator is executed at every sampling point. For such position-invariant processes, an arrayed structure is very useful. Many sampling data are inputted to many processing elements at a time. Then all processing elements execute the same operation simultaneously. Increasing the number of processing elements by a certain factor should cause the processing time to decrease by the same factor.

One cause reducing the effect of an arrayed structure is the occurrence of irregular flow such as conditional jump. In such cases, all processing elements can not be controlled by one sequence. Some PEs will have to be in a halt state. To avoid this problem and increase efficiency of parallel processing, a special micro-processor was developed which can absorb irregularities and do such special calculations as summing products, table look up and getting hamming-distance at high speed.

It is difficult to control the array. Therefore a usage of micro-instructions was planned to control executions in a processing element and data's flows between the array and a picture memory.

### 3. SYSTEM OUTLINE

A block diagram of our IP is shown in Fig. 1. It consists of a parallel processing part and the other part. The former comprises a processing element array (PE Array), a picture memory (PM), local registers (LR), a micro-instruction control (MC) and input/output control (IOC).

Image data are inputted through the IOC and stored in the PM. They are originally obtained 6 bits in the IOC. The image data are then transmitted to the PE Array through the LR. The LR can store a  $4 \times 4$  local-image or an  $8 \times 8$  one. The PE Array processes it parallelly, and the results are stored again in the PM.

The PM is a memory for images consisting of three banks of  $256 \times 256$  pixels. The access to the PM is controlled by micro-instructions in a program. The accessing unit is a  $4 \times 4$  square with the upper-left corner address ( $4n, 4m$ ), or one pixel. The letters  $n$  and  $m$  stand for integer from 0 to 63.

The PE Array consists of sixteen processing element (PE). Each PE is a special micro-processor comprising a controller, which interprets commands and controls calculations in synchronization with an external clock. The commands comprise addition, subtraction, multiplication, summing products, logical operation,

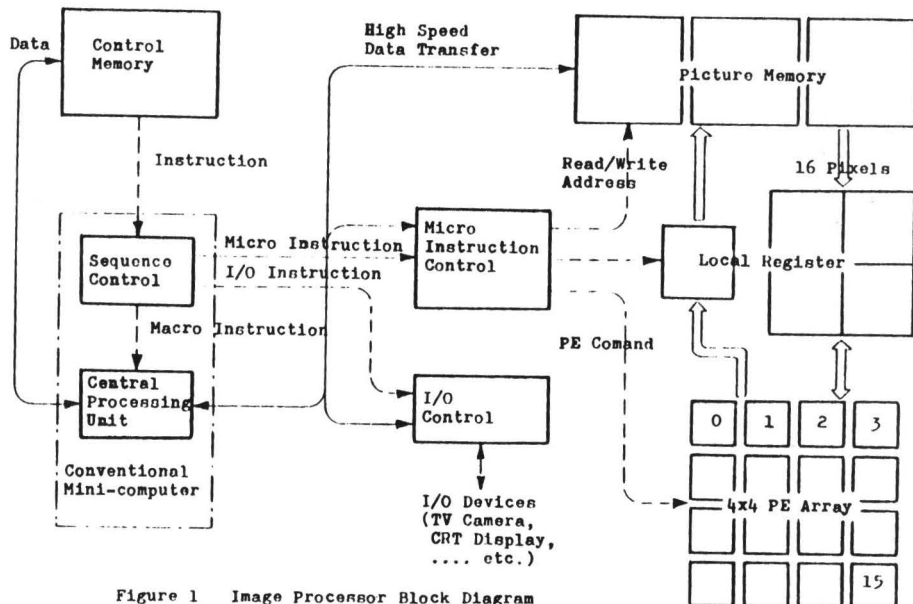


Figure 1 Image Processor Block Diagram

table look up and so on. Each PE has a 256 words 16 bits scratch pad memory. The scratch pad memory can store a sequence of PE commands. Therefore each PE can execute its own program. The depth of calculations is twelve bits long. The execution time is  $0.35 \mu\text{S}$  and the cycle time of the PM is  $0.7 \mu\text{S}$ .

The LR has shift functions in an  $8 \times 8$  local image. The PE Array can obtain the free  $4 \times 4$  square pixels from the PM by these functions. Also, PEs can communicate each other through the LR.

#### 4. BASIC OPERATIONS

In order to describe functions in the PE Array, several notational definitions are given as follows;

$D(x,y)$ ; a value of pixel  $(x,y)$  in the original image.

$$D(x,y) = \begin{bmatrix} D(x,y), & \dots, & D(x+4,y) \\ \vdots & & \vdots \\ D(x,y+4), & \dots, & D(x+4,y+4) \end{bmatrix}$$

$R(x,y)$ ; a result by a certain operation in a PE of the PE Array.

$F(), G()$ ; functions executed by a combination of PE's commands.

In the PE Array, sixteen PEs process local images in the LR simultaneously. Therefore sixteen results  $R(x,y)$ ;

$$R(x,y) = \begin{bmatrix} R(x,y), & \dots, & R(x+3,y) \\ \vdots & & \vdots \\ R(x,y+3), & \dots, & R(x+3,y+3) \end{bmatrix}$$

are obtained simultaneously.

The representative basic operations in the PE Array are as follows.

(a) Data conversion.

$$R(x,y) = F(D(x,y))$$

This is used for thresholding, contrast stretching and so on. In a thresholding of a  $256 \times 256$  image, it takes about 11.5 mS.

(b) Local operation.

$$R(x,y) = F(D(x,y))$$

For example, in a case of two-dimensional filterings the results are as follows.

$$R(x+m,y+n) = \sum_{i=0}^4 \sum_{j=0}^4 f(i,j) \times D(x+i,m,y+j+n)$$

$0 \leq m \leq 3; \quad 0 \leq n \leq 3$

where  $f(i,j)$ s are weight coefficients.

In all PEs, twenty-five times' executions of the PE command "SUM OF PRODUCT" are done. It takes 34.4 mS in a  $256 \times 256$  image. By this local operation, spacial differentiation, noise elimination, template matching, and so on are available.

(c) Sequential processing.

$$R(x,y) = F(D(x,y), Q(x-s,y-t))$$

$$Q(x,y) = G(D(x,y), Q(x-s,y-t))$$

Where  $Q$  is PE's internal states

specified by several flags and contents of the scratch pad memory. The point  $(x-s,y-t)$  shows a preceding processing point to point  $(x,y)$ . Values of letter  $s$  and  $t$  depend on a scanning in the PM.

This calculation is used for histogram's calculation, area counting and so on. For example, the processing speed of obtaining pixel value's distributions is 28 mS.

(d) Manipulation between two images.

$$R(x,y) = F(D_1(x,y), D_2(x,y))$$

where  $D_1(x,y)$  and  $D_2(x,y)$  show values of different images' pixels.

For example, it takes about 18 mS to get an add-image or exclusive OR-image from two images.

#### 5. CONCLUSION

An image processor based on an arrayed structure was implemented. It deals with images consisting of  $256 \times 256$  pixels and processes them by a  $4 \times 4$  processing elements' array. It was constructed in 1977 and used to study processor's architecture for the object recognition. By these studies, it has been established that the array of our IP is useful for high speed processing.

The basic operations in the array are data conversions, local operations, sequential processings, manipulations between two images, and so on. By them, many kinds of image processing can be done in this array.

#### REFERENCES

- [1] M. J. E. Golay, "Hexagonal Parallel Pattern Transformations", IEEE Trans. Comput., Vol. C-18, pp. 733-740, Aug. 1969.
- [2] K. S. Fu, "Special Computer Architectures for Pattern Recognition and Image Processing An Overview", Proc. AFIPS, Vol. 57, pp. 1003-1013, June 1978.