

# A Data Flow Digital Image Processing System and Its Real-Time Morphological Algorithms

Xu Ning, Ding Xiaoqing  
Dept. of Electronic Engineering  
Tsinghua University, Beijing, China

## Abstract

In this paper, the concept of data flow machine is briefly described. And then, a high speed image processing system based on data flow ImPPs (Image Pipelined Processor) is given and analysed. Some examples are also given to demonstrate the data flow concept and to indicate the applications of this system. Finally, based on the idea of morphological spectrum, a decomposition method of arbitrary morphological structuring elements is presented to expand the applications of this system.

## 1- Introduction

Image processing and pattern recognition are important research areas and have varied applications. But in most cases, the digitized image arrays are too large to be processed efficiently in a small computer. To solve this problem, an image processing system is usually needed to provide adequate bulk storage and efficient processing capabilities.

With regard to the varied operations of image processing and pattern recognition, image processing systems are generally bulky and expensive [1]. In order to improve the computational capabilities of conventional computers, many architectures such as parallel and pipeline architecture are introduced. A high speed system with low cost is important for applications. Now, with rapidly developing VLSI techniques, many advanced small processing systems are being developed [2].

Data flow architecture is a new concept which has many advantages over the conventional architectures. Based on NEC data flow Image Pipelined Processor (ImPP)  $\mu$ PD7281, we have developed a high speed digital image processing system. Its block diagram is shown in Figure 1.

The host computer of this system is IBM PC/AT or their compatible personal computer. The whole system has four main parts: input/output interface circuit, image memory, image processors, real-time processing circuit.

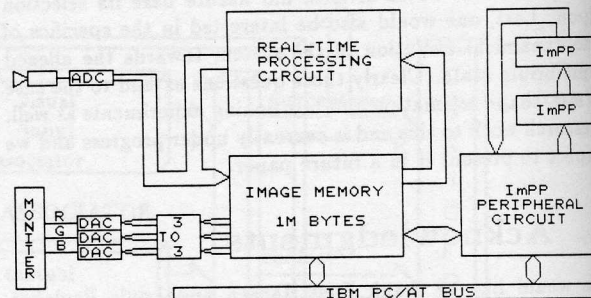


Figure 1. System Block Diagram

## 2- Data-Flow Concept

The operations of data flow machines are identified by 'tokens'. Each token consists of a tag and a data. The tag informs the processor what is to be done to data. When all the tokens needed for a particular operation are matched, the operation can be executed. So in a data flow machine, there is no predetermined order of events. There is also no need of time for data flow processors to fetch instructions, load data and store the intermediate results of the calculation [3] [5].

For example, we want to perform  $Y = A * B + C * D$ . We only need input four tokens, each of which contains a data and a tag indicating what operation and with which data this operation should be performed. First, when token A meets token B, data A is multiplied by data B and that generates token E. The data E in token E equals  $A * B$ . Secondly, when token C meets data D, the token F is generated. Finally, the token E and token F are performed, the output token Y is generated and sent to the host computer or the next processor. The flow graph is shown in Fig.2(a). On the contrary, the same simple operations in von Neumann machines will be time consuming.(Fig.2(b))

Generally, the difference of executing data between data flow machine and von Neumann computer can be illustrated by Fig.3.

The another advantage of data flow machine is its facilitated multiprocessing operations. With tokens,

processors are dynamically connected as a network, in which each processor's loads or tasks are scheduled by tokens. This dynamic load scheduling is intrinsic to a data flow machine.[5]

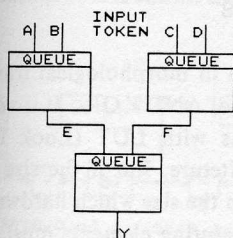


Fig. 2(a) Flow Chart

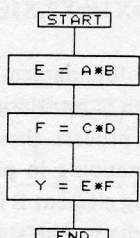


Fig. 2(b) Flow Chart

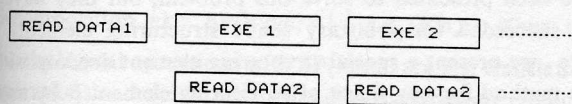


Fig. 3(a) Data-Flow Machine Executing Flow Graph

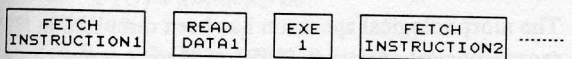


Fig. 3(b) Von Neumann Computer Exe. Flow Graph

### 3- System Architecture

The whole architecture of this system is designed especially for image processing and pattern recognition. The digital signal processor in this system is NEC ImPP. Through peripheral circuit, these ImPPs are connected to host computer and image memory. ImPPs can read and write image memory directly.

The real-time processing circuit is another important part of this system. It and the input/output interface circuit of the memory compose a real-time feed-back processing circuit in which some TV rate processing task can be performed.

Now, some main features of this image system are given.

#### (1) Data flow architecture ImPP

The ImPP  $\mu$ PD7281 is the first implementation of data flow architecture on a circular pipeline in a monolithic integrated circuit. It is well suited for image processing such as two-dimensional convolution, shrinking and rotation. Several ImPPs can be easily cascaded to form a multi-processor system. The ImPP which is working on the data is defined by the tag in the token. If the module

number in the tag does not match that of ImPP, the token will go to next ImPP directly and will not be processed by the circular pipeline inside the ImPP. Therefore, the pipeline bottle-neck can be overcome in this system if every ImPP's task is arranged efficiently.

In this given system, only two ImPPs are used. Considering that more ImPPs may be needed for certain requirements, some hardware for setting more module number is also included in this system. So just by adding some ImPP chips, the source of processors is expanded.

#### (2) Efficient image memory

Image memory is the central unit of image processing systems. As a image memory, it must be efficient and convenient for processors to get data or write the results.

In one TV frame time, only few memory cycles of this system are needed for displaying and refreshing memory. Almost all cycles are for host or ImPPs access with display undisturbed. The image memory is mapped into host addressing space when the memory is for host access. Host computer can read or write image memory just as reading or writing its internal memory. ImPPs can visit the whole space of image memory directly.

#### (3) Multiple function system

The system has been designed for multiple uses. The size of memory in this system can be arranged as four  $512 \times 512 \times 8$  bits or  $2048 \times 512 \times 8$  bits or  $1024 \times 1024 \times 8$  bits. All these are programmable. Four  $512 \times 512 \times 8$  bits mode is standard video mode;  $2048 \times 512 \times 8$  bits mode is for scanner input cases;  $1024 \times 1024 \times 8$  bits mode is for high-resolution image data.

#### (4) Real-time processing circuit

This circuit has two parts. One part consists of an ALU, which takes logical processing between two frames in the memory. The other part is a special architecture and its principle diagram is shown in Figure 4. With different contents of the first and second LUT, many processing functions which require access to a local neighbourhood of each pixel can be conducted in this circuit. Considering the limitation of LUT size, here we only process binary image data. With different contents of LUT1, varied functions can be conducted. LUT2 synthesizes several LUT1's results and gives an overall output.

A typical application of this circuit is mathematical morphology operations for image analysis. We use

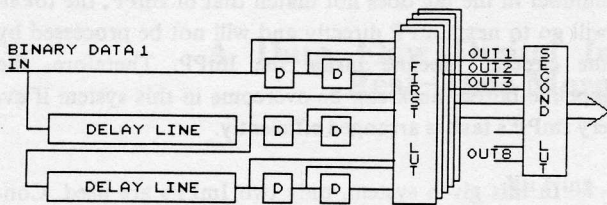


Figure 4. Real-Time Processing Circuit

mathematical morphology for stroke extracting of chinese characters. For example, Mask A (Fig. 5) is used for extracting horizontal strokes. Mask B (Fig.5) is used for extracting vertical strokes. When input data meet the mask, the output of LUT1 is '1'. Otherwise the output is '0'. The mask c is used in LUT2 for determining where a horizontal and a vertical stroke across, i.e., where there is a crossing point. LUT2 encodes the features of a chinese character and outs this result to image memory for recognition.

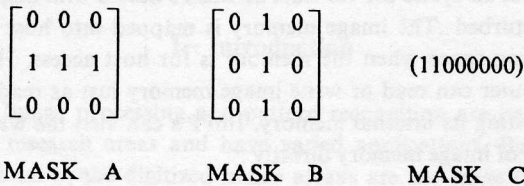


Figure 5 Masks for stroke extraction

some operations which use large structuring elements must be divided into small ones and then can be executed in this circuit. Because the number of delay units in each line is also programmable, this circuit is very flexible for different operations.

#### 4- Morphological Operations With Decomposition Method of Large Structuring Element

First, let's review some main morphological operations:

$$\text{Erosion: } X \ominus B = \{a: Ba \subseteq X\} = \bigcap_{b \in B} X_b$$

$$\text{Dilation: } X \oplus B = \{a: Ba \cap X \neq \Phi\} = \bigcup_{b \in B} X_b$$

$$\text{Opening: } X \circ B = (X \ominus B) \oplus B$$

$$\text{Closing: } X \bullet B = (X \oplus B) \ominus B$$

$$\text{Translation: } X_b = \{a + b, a \in X\}$$

X is the original image data, B is the structuring element. There are some basic characters about the operations:

- (A)  $B1 \oplus B2 = B2 \oplus B1$   
 $B1 \oplus (B2 \oplus B3) = (B1 \oplus B2) \oplus B3$   
 $(B1 \cup B2) \oplus B3 = (B1 \oplus B3) \cup (B2 \oplus B3)$   
 $B1 \ominus (B2 \oplus B3) = B1 \ominus B2 \ominus B3$
- (B)  $\Rightarrow X1 \neq X2, X1 \ominus B = X2 \ominus B$   
 $\Rightarrow X1 \neq X2, X1 \oplus B = X2 \oplus B$

The fundamental operation in morphological transformation is translation and logical AND / OR. It is very easy to process these operations with LUT (Look Up Table) mapping in pipeline architecture. But if the size of structuring element B is larger than the size which hardware can handle in one stage, the structuring elements must be decomposed into smaller ones. Structuring element decomposition is an essential and difficult task. A few papers have been presented to solve this problem, but they have not succeeded for arbitrary shape structuring elements. Here, we present a special structuring element decomposition method for any shape of structuring element B by using the idea of morphological spectrum.

The morphological spectrum has been discussed in [7]. The morphological spectrum SP(B) of B with respect to a primitive element e is defined as:

$$SP(B) = \bigcup_n SP(B, n, e)$$

$$SP(B, n, e) = B \ominus ne - B \ominus neoe \quad (E4.1)$$

$$B = \bigcup_n SP(B, n, e) \oplus ne \quad (E4.2)$$

$$ne = (e \oplus e \oplus e \dots \oplus e)$$

$$\min\{k | B \ominus ke \neq \Phi\} < n < \max\{k | B \ominus ke \neq \Phi\}$$

where SP(B, n, e) is the n-th morphological spectrum with respect to primitive element e and the primitive element e is chosen to be small enough to be performed by the special pipeline circuit.

It has been proven in reference [7] that any arbitrary shape image B can be reconstructed by its morphological spectrum with respect to the primitive element e as the equation (E 4.2). By using the morphological spectrum, the

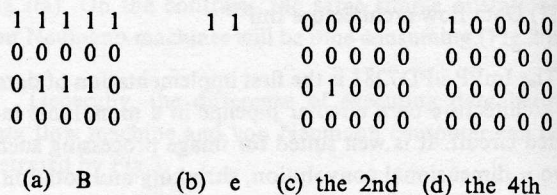


Fig. 6.1 Morphological Spectrum Example

image data can be compressed heavily. For example , the image B is shown as Fig 6.1(a) and the primitive element e is shown as Fig.6.1(b). Then, the spectrum of B will only have the 2nd and 4th spectrum as Fig 6.1(c) and (d) respectively. The diagram of getting the morphological spectrum of B is shown in Fig.7.

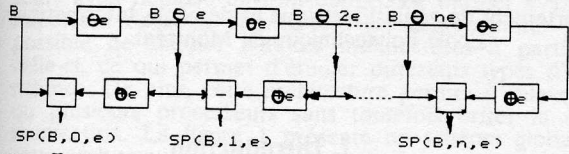


Fig. 7 Diagram of Morphological Spectrum Algorithm

If  $SP(B,n,e)$  has more than one elements, it can also be decomposition as :  $SP(B,n,e) = \bigcup_j SP_j(B,n,e)$ . Thus the morphological operations with  $SP(B,n,e)$  are just translation operations and can be easily handled by hardware circuit.

Then  $B = \bigcup_n \bigcup_j SP_j(B,n,e) \oplus ne$   
 And Dilation:

$$X \oplus B = X \oplus (\bigcup_n \bigcup_j SP_j(B,n,e) \oplus ne)$$

$$= \bigcup_n \bigcup_j X \oplus ne \oplus SP_j(B,n,e)$$

Erosion:

$$X \ominus B = X \ominus (\bigcup_n \bigcup_j SP_j(B,n,e) \oplus ne)$$

$$= \bigcap_n \bigcap_j X \ominus ne \ominus SP_j(B,n,e)$$

Here , the morphological operations with respect to e and  $SP_j(B,n,e)$  can be performed in the special real-time processing circuit , so the operations with respect to any structuring element B can be implemented by decomposed structuring element. The fast algorithms for dilation and erosion with respect to B by using decomposition method are shown in Fig.8.

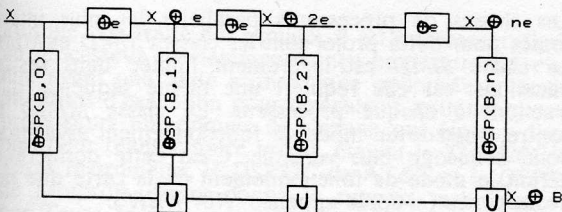


Fig.8(a) Dilation with Decomposed Structuring Element

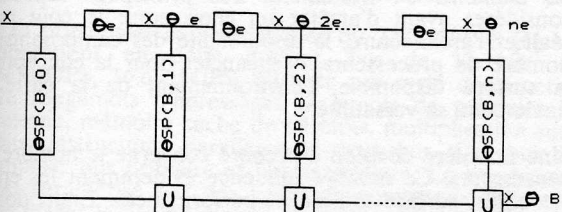


Fig.8(b) Erosion with Decomposed Structuring Element

### 5- Conclusion

The image system described in this paper is based on data flow architecture processors. The whole system is integrated into two PC boards which can be plugged into I/O slots of IBM PC / AT or their compatible computers. it is designed especially for image processing and pattern recognition.

Since image data is very large, real-time processing unit is necessary for image processing task. Due to LUT size, only binary image data can be processed in this circuit. How to process grey image data without increasing circuit hardware largely is worthy of further consideration.

### 6- References

- [1] KITTLER J. and DUFF M.J.B. 'Image Processing System Architecture', Research Studies Press LTD
- [2] 花木真一, 岩下正雄, 寺嶋廣克, '画像処理システム事例', 'テレビジョン' 学会誌, Vol.41 No.2 ,pp189-197,1987
- [3] NEC 'μPD7281 エーサース'. マニュアル, December, 1986
- [4] NEC 'μPD9305 エーサース'. マニュアル, October, 1986
- [5] COLLETT R. 'CPU Architecture Part II: Dataflow Computers Encroach On von Neumann Territory', DIGITAL DESIGN, 1984
- [6] ABBOTT L., HARALICK R.M., ZHUANG X.H. 'Pipeline Architecture for Morphologic Image Analysis', Machine Vision and Applications, No.1, 1988
- [7] Ding Xiaoqing, 'Morphological Spectrum Image Analysis and Shape Representation.' ACTA Electronica Sinica Vol.17 No. 6 1989, PP47-54.