

A Study on Parallel Processing System for Automatic Segmentation of Moving Object in Image Sequences

Hyung Lee, Jong Won Park*

Department of Computer Engineering

*Department of Information Communications Engineering
Chungnam National University

220 Gung-Dong Yusong-Gu, Taejon, 305-764, KOREA

Tel: +82-42-821-7793, Fax: +82-42-823-5586

E-mail: hyung@crow.cnu.ac.kr

Abstract: The new MPEG-4 video coding standard enables content-based functionalities. In order to support the philosophy of the MPEG-4 visual standard, each frame of video sequences should be represented in terms of video object planes (VOP's). In other words, video objects to be encoded in still pictures or video sequences should be prepared before the encoding process starts. Therefore, it requires a prior decomposition of sequences into VOP's so that each VOP represents a moving object. A parallel processing system is required an automatic segmentation to be processed in real-time, because an automatic segmentation is time consuming.

This paper addresses the parallel processing system for an automatic segmentation for separating moving object from the background in image sequences. The proposed parallel processing system comprises of processing elements (PE's) and a multi-access memory system (MAMS). Multi-access memory system is a memory controller to perform parallel memory access with the variety of types: horizontal, vertical, and block access way. In order to realize these ways, a multi-access memory system consists of a memory module selection module, data routing modules, and an address calculation and routing module. The proposed system is simulated and evaluated by the CADENCE Verilog-XL hardware simulation package.

1. Introduction

Unfortunately, image segmentation is recognized as an ill-posed problem and still remains unsolved. In addition, MPEG-4 assumes that VOP's in video are available prior to the encoding process. Therefore, real-time generation of the VOP's is a very attractive problem, and the accurate segmentation of high quality is required for high-level image analysis such as object recognition, image understanding, and scene interpretation and is necessary for authoring multimedia content, etc. So, a parallel processing system is necessary for an automatic segmentation to be processed in real time. In order to speed up this application attached special purpose parallel processing system, which can be communicated with a host computer via a fast PCI local bus system, is required.

We have been developing a parallel processing system to improve processing speed during image applications, using an architecture similar to SIMD. The system consists of a processing unit; a control memory which stores instructions and common data; DMA controller which has set of registers to fetch instructions from the control memory and issues it to PE's; n PE's

which interpret and execute instructions synchronously; MAMS which provides data elements to the PE's simultaneously; and m external memory modules which store data elements to be manipulated. Several authors considered multi-access memory systems[1,2,3,4,5]. In particular, Parks and Harper[5] proposed a memory system for the SIMD construction of a Gaussian pyramid, where the number of memory modules of the memory system were 2^n and $2^n + 1$ respectively.

In this paper, we propose a parallel processing system involving a multi-access memory system for a simultaneous access to the data elements within three access types with a constant interval. Also, parallel method for an automatic segmentation of moving object in image sequences to be applied to the system is suggested.

This paper is organized as follows. A parallel processing system and a multi-access memory system are detailed in section 2 and section 3, respectively. Section 4 presents the automatic segmentation developed ETRI and considerations for applying to the system. Section 5 presents experimental results and finally the conclusion and discussion are presented in section 6 followed by the references.

2. Parallel Processing System

If an algorithm consists of many identical operation on different image points, an SIMD architecture in which one processor unit operates N PE's synchronously will be adequate to the algorithm. If an algorithm needs to access data in any direction with an interval corresponding to the length of between data, a multi-access memory system, which provides simultaneous access to N data elements with access types, will be adequate to the algorithm. The type may be a block, a horizontal or a vertical with some interval. Their properties are involved in some image processing algorithm or some parts of an algorithm.

The parallel processing system we proposed consists of a processing unit (i960VH Embedded-PCI Processor); a local memory which stores instructions and common data; DMA controller which has set of registers to fetch instructions from the local memory and issue them to PE's; n PE's which interpret and execute instructions synchronously; MAMC which provides data elements to the PE's simultaneously; and m external memory modules which store data elements to be manipulated.

The processor unit (PU) controls the system and communicates with host computer via PCI bus. DMA controller fetches an instruction and stores it in a register

pool also synchronously transfers data or an instruction to PE's and controls them. PE is designed as ALU with the functionality that is interpreting an issued instruction. And, to perform an application, DMA controller steals bus cycles until the end of the application. PE can execute two kinds of instructions: memory-reference instructions for accessing m external memory modules via MAMC and 16 general instructions including register-reference instructions and I/O instructions. Therefore, an application to be processed on the system is compiled to operation codes within 18 instructions. And, when each of two instructions is in different set of instructions, they are executed at the same time. That is, one of memory-reference instructions and one of general instructions are executed simultaneously. Hence, a memory-reference instruction followed by a general instruction is executed in a memory access cycle and vice versa. It is reducing processing time in some modules, for example, convolution mask operations frequently used in spatial domain.

The system provides logically two-dimensional addressing mode that is used in (r, c) -based image domain. Therefore, most of image processing in spatial domain is done with enough parallel processing power. The block diagram of the system is presented in Figure 1.

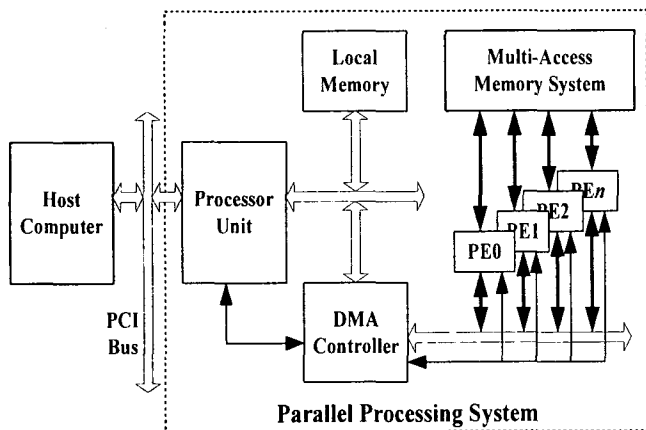


Figure 1. The block diagram of the proposed parallel processing system, where is $n = 4$

3. Multi-access Memory System

For a parallel processing system with n PE's, it is necessary to use a multi-access memory system to reduce the memory access time. Also, The memory system has the important goals to provide the efficient utilization for PE's of the parallel processing system we proposed. The goals are as follows: various access types and constant interval between the data elements, simultaneous access with no restriction on the location, simple and fast address calculation and routing circuitry, and small number of memory modules.

The memory system consists of a memory module selection circuitry, a data routing circuitry for WRITE, an address and a routing circuitry, m memory modules, and a data routing circuitry for READ. In order to distribute the

data elements of the $M \times N$ array $I(*,*)$ among m memory modules, a memory module assignment function must place in distinct memory modules array elements that are to be accessed simultaneously. Also, an address assignment function must allocate different addresses to array elements assigned to the same memory module [3].

An MAMC, comprised of a memory module selection module, a data routing circuitry, and an address and a routing module, is implemented to the pipelined because the parallel processing system is pipelined. Therefore, in the case of sequential memory operations, memory access times are reduced in comparing with that of the original. The block diagram of the pipelined multi-access memory system is presented in Figure 2.

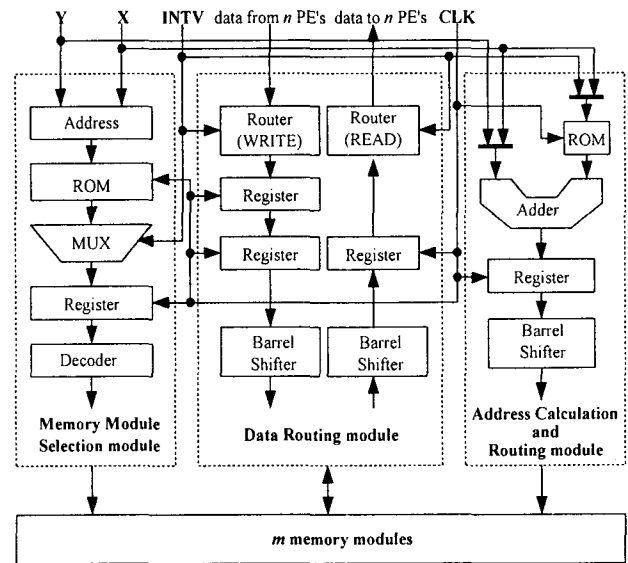


Figure 2. The block diagram of the pipelined multi-access memory system

4. Segmentation of moving object in image sequences

We investigated the segmentation methods of moving object in video, using the one developed by ETRI (Electronics and Telecommunications Research Institute, KOREA), whose performance is comparatively accurate [6,7,8].

The method utilizes spatio-temporal information. 1) For localization of moving objects in the image sequence, two consecutive image frames in the temporal direction are examined and a hypothesis testing is performed by comparing two variance estimates from two consecutive difference images, which results in an F -test. 2) Spatial segmentation is performed to divide each image into semantic region and to find precise object boundaries of the moving objects. The temporal segmentation yields a change detection mask that indicates moving areas (foreground) and nonmoving areas (background), and spatial segmentation produces spatial segmentation masks. A combination of the spatial and temporal segmentation masks produces VOP's. Although the method showed reasonable performance, one of the most serious drawbacks

is the time consuming processing.

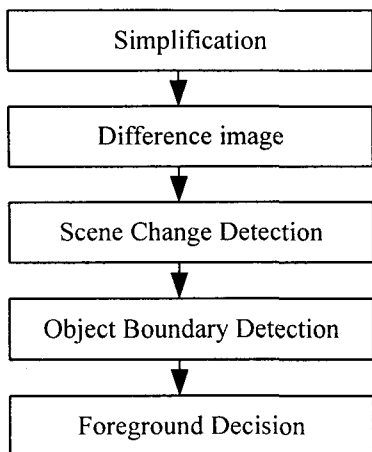


Figure 3. The flow diagram of automatic segmentation

We tried to transform serial modules composing the automatic segmentation into parallel ones in step by step. In the transformation procedure, some modules in the application were transformed with easy and transformed modules operating in the same functionality as old modules. But, any modules don't seem to be transformed because any identical memory access type doesn't be detected through all of its portions. In other words, it is possible to transform the module for the parallel processing system, but processing speed of transformed module was insignificant compared to the original. In this case, new modules were considered by using different methods. In spite of those procedures to make modules for the system, some of modules were serially processed on the host because of their lower degree of parallelism.

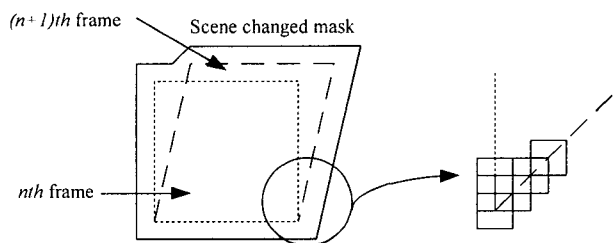


Figure 4. Flooding section using block access with interval 1 on the proposed parallel processing system

Figure 3 depicts a diagram of the proposed method. In the simplification step, morphological filters are used for the purpose of simplifying the image to make easier the image segmentation in ETRI's method. But, the module needs longer processing time than other modules because the filters are recognized as a time-consuming work. Median filter was used, instead of morphological filters, which yielded the acceptable results and was adequate to the system because of convolution mask operation to be carried out with block access way. And, a difference image was simply obtained by block access way. In the scene change detection, a scene changed region was detected by a

mathematical statistics, that is, a region is recognized as a scene changed one when the covariance of the region and the global region is less than a constant value that resulted from a lot of experiments. A watershed section to find object boundary consists of flooding section and new minima detection. Flooding section using the block access way is illustrated with a simple graphical depiction in Figure 4. New minima detection was processed on the host. Finally, Foreground was simply detected from the previous step.

5. Experimental Results

The segmentation on the parallel processing system described had been experimentally investigated by means of computer simulations. One test sequence, MOTHER_DAUGHTER with the QCIF format (176x144 pixel elements) had been used with a reduced frame frequency of 10Hz. Figure 5 shows a series of the results for the sequence during the process on the proposed system. Notice that results depicted in Figure 5 were not through a post-processing.

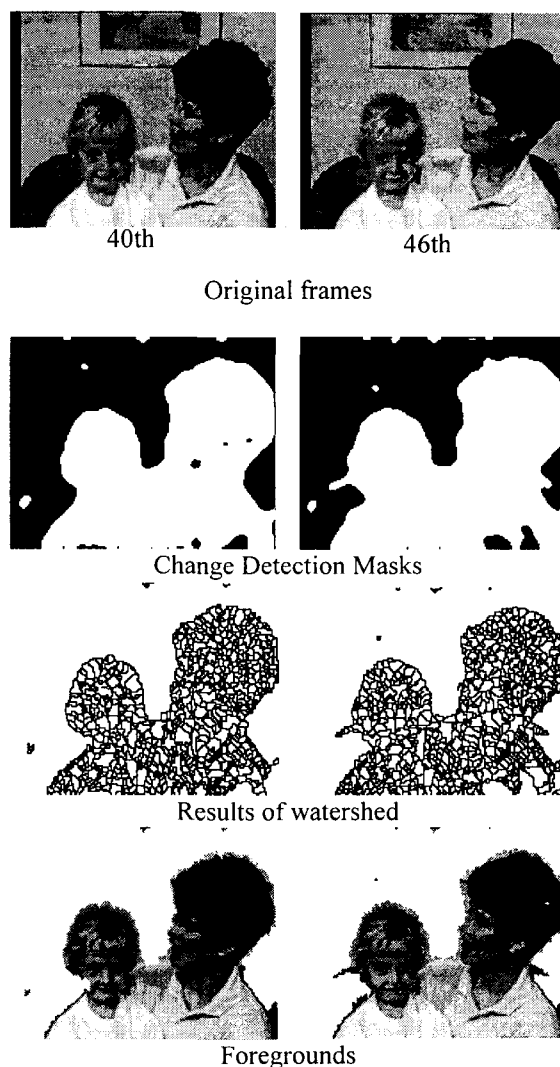


Figure 5. Results during processing automatic segmentation on the parallel processing system

The proposed system was verified by CADENCE Verilog-XL hardware simulation package and processing speed of the segmentation on the system was 1.5 times faster than the original method, where the number of PE's was 4. But the value 1.5 was achieved, it is just estimated value because the proposed system has not yet been manufactured into a circuit board. If large number of PE is equipped with the system, it can be speculated that the speedup performance be pushed up to the limitation predicted by Amdahl's Law. Figure 6 presents a wave generated while the system is simulated by the simulation package.

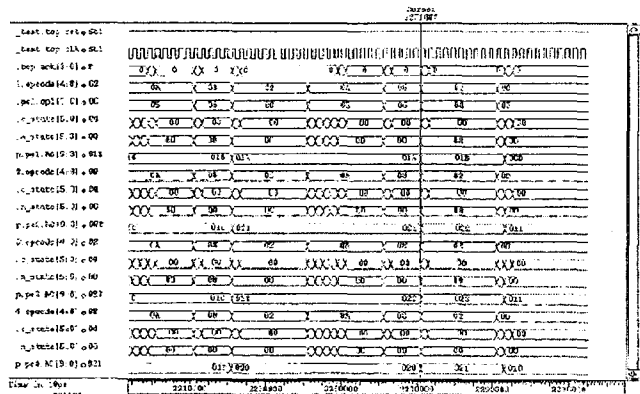


Figure 6. A wave generated while the system is in pre-layout simulation by CADENCE Verilog-XL.

6. Conclusions

The demands for processing multimedia data in real-time using unified and scalable architecture are ever increasing with the proliferation of multimedia applications. We presented a parallel processing system to achieve the demands for speedups and applied to automatic segmentation for the support of an object-based coding standard MPEG-4, which was recognized time-consuming work. It was confirmed that the system had been able to support enough scalability to the application and improved its processing speed in real-time processing. If large number of PE is equipped with the system, it can be speculated that the speedup performance be pushed up to the limitation predicted by Amdahl's Law.

Although the comparison value previously mentioned was obtained through simulation and speedup performance during the application on the system was achieved, it was just estimated value because the proposed system has not yet been manufactured into a circuit board.

Unfortunately, some problems occurred in transferring a lot of data elements from the host to the system and vice versa during processing the application on the system. That is, the time for transferring data allocated more than the processing time. To solve this, the bus bandwidth needs to be improved on the system side and new specific methods to the system be developed on the method side.

Acknowledgement

This work was supported by Ministry of Information and

Communications of Korean government under the project, "Parallel Processor Architecture for Automatic Segmentation of MPEG-4".

References

- [1] P. Budnik and D. J. Kuck, "The organization and use of parallel memories," *IEEE Trans. Comput.*, Vol. C-20, pp. 1566-1569, Dec. 1971.
- [2] D. C. Van Voorhis and T. H. Morrin, "Memory systems for image processing," *IEEE Trans. Comput.*, vol. C-27, pp. 113-125, Feb. 1986.
- [3] Jong Won Park, "An efficient memory system for image processing," *IEEE Transactions on Computers*, vol. C-35, no. 7, pp. 33-39, 1986.
- [4] D. H. Lawrie and C. R. Vora, "The prime memory system for array access," *IEEE Trans. Comput.*, vol. C-31, pp. 435-442, May 1982.
- [5] J. W. Park and D. T. Harper II, "An Efficient memory system for the construction of a Gaussian pyramid," *IEEE Trans. Parallel and Distributed Systems*, vol. 7, no. 8, pp. 855-860, Aug., 1996.
- [6] Jae Gark Choi, Munchurl Kim, Myoung Ho Lee, and Chieteuk Ahn, "Automatic Segmentation based on spatio-temporal information," *ISO/IEC JTC1/SC29/WG11 MPEG97/m2091*, Bristol, April 1997.
- [7] Jae Gark Choi, Si-Woong Lee, and Seong-Dae Kim, "Spatio-temporal Video Segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 279-286, April 1997.
- [8] Munchurl Kim, J. G. Choi, D. H. Kim, H. Lee, M. H. Lee, C. Ahn, and Y. Ho, "A VOP Generation Tool: Automatic Segmentation of Moving Objects in Image Sequences Based on Spatio-Temporal Information," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1216-1226, Nov., 1999.