

A HIGH-SPEED VISION SYSTEM FOR MOMENT-BASED ANALYSIS OF NUMEROUS OBJECTS

Yoshihiro WATANABE, Takashi KOMURO, and Masatoshi ISHIKAWA

Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.
E-mail: Yoshihiro_Watanabe@ipc.i.u-tokyo.ac.jp

ABSTRACT

We describe a high-speed vision system for real-time applications, which is capable of processing visual information at a frame rate of 1 kfps, including both imaging and processing. Our system performs moment-based analysis of numerous objects. Moments are useful values providing information about geometric features and invariant features with respect to image-plane transformations. In addition, the simultaneous observation of numerous objects allows recognition of various complex phenomena. The proposed system achieves high-speed image processing by providing a dedicated massively parallel co-processor for moment extraction. The co-processor has a high-performance core based on a pixel-parallel and object-parallel calculation method. We constructed a prototype system and evaluated its performance. We present results obtained in actual operation.

Index Terms— Moment, Real-time image processing, Vision system, Massively parallel processor, High-frame-rate imaging

1. INTRODUCTION

High-speed image processing at high frame rates is highly effective in real-time applications, including automation, inspection, robot control, microscope observation, and man-machine interfaces. However, the performance demonstrated thus far is not high enough to observe and control phenomena that typically require data output at high throughput rates and low latency. For example, machine control requires kilohertz-order responses, which is significantly higher than the frame rates achieved to date by conventional systems, specifically the video rate used in the NTSC system (30 Hz).

As a novel approach for such applications, we propose a new high-speed vision system in this paper. In particular, our system can be employed in applications using moment-based analysis. We detail the system architecture and show its performance. We demonstrated real-time extraction of moment information from more than 1,000 regions at a frame rate of 955 fps. We also present results of evaluation experiments.

2. MOMENT-BASED ANALYSIS OF NUMEROUS OBJECTS

In this paper, moments mean geometric moments. The $(i+j)$ -th moment is calculated from:

$$m_{ij} = \sum_x \sum_y x^i y^j I(x, y), \quad (1)$$

where $I(x, y)$ is the value of a binary image at pixel (x, y) . The moment can be converted to various geometric features, such as size, centroid, orientation, rough shape information, and so on [1]. Also, the moment can express values that are invariant with respect to image-plane transformations. These advantages allow novel approaches for object identification, image measurement, and pattern recognition.

The applications that we focus on in this paper include the real-time image analysis of more than 1000 segmented regions. This analysis enables observation of various local changes in patterns and increases the amount of information obtained from the analysis. This allows improved accuracy and flexibility in application development, compared to obtaining only moments of an entire image that is not divided into segmented regions. For example, fluid measurement typically requires acquiring the motion of over 1,000 particles at 1 kfps to measure complex flow conditions. However, the moment-based analysis of such a large number of objects involves a large amount of calculation and is difficult to perform in real-time.

3. OVERVIEW OF HIGH-SPEED VISION SYSTEM FOR MOMENT-BASED ANALYSIS

First, we describe the overall architecture of our high-frame-rate imaging and high-speed processing system. In related work, this type of system was designed to avoid an image transfer bottleneck between sensors and processor. One approach is to use a focal plane processor that integrates a sensor array and a processor array in one chip to remove the data transfer step [2]. Another approach is to read out and process only the surrounding image of a certain object [3]. Although the maximum frame rates of these approaches are quite high, the systems were quite specialized, and it was therefore difficult to extend their functionality and ensure processing flexibility.

On the other hand, parallel A/D conversion and high-speed image transfer based on Low Voltage Differential Signaling (LVDS) has allowed imaging at high frame rates, as high as 1 kfps. However, such sensors will not find practical uses in real-time applications until they incorporate high-speed image processing. Therefore, in our present work, the first problem that we focus on is a suitable system architecture for performing a large amount of processing with high throughput.

Our proposed system involves three effective strategies: (A) parallelization of algorithms to minimize the amount of calculation, (B) optimal processor circuit structure for the algorithms, and (C) large scalability. We realize these features by introducing a swappable massively parallel processing module as a co-processor having specialized functionality and configuration.

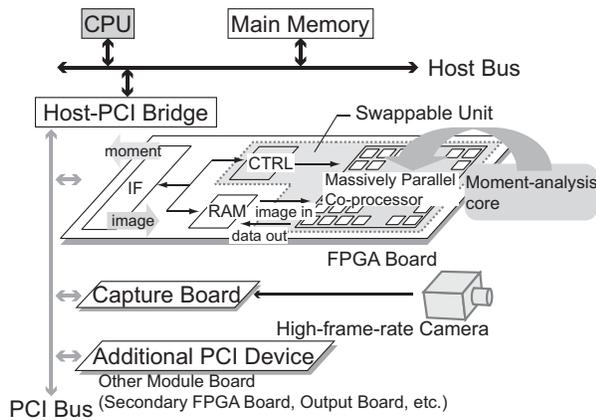


Fig. 1. System architecture.

Fig.1 shows the architecture of the overall system. It consists of a PC, a high-frame-rate camera, and a co-processor integrating image-processing cores. The co-processor is implemented in a Field Programmable Gate Array (FPGA) and is capable of dedicated parallel operations. The functionality of this type of system can be reconfigured by swapping the co-processor or changing the device setup, thus providing high-performance processing and high scalability in various application fields. Based on this architecture, we configured the system as a high-speed vision system for moment-based analysis of numerous objects by providing a co-processor with such functionality. This is merely the configuration we used in the present study, however. The system may be reconfigured for other tasks by installing a co-processor with suitable functionality.

The FPGA board we used for implementing the co-processor was KAC-02A made by Mitsubishi Electric Engineering Company. Two FPGA chips (XC2V6000) for general-purpose computations were mounted on this board. Each contained 67,584 four-input look up tables (LUTs) and 67,584 flip flops (FFs). The co-processor unit was formed of the cores in the XC2V6000 chips. The co-processor communicated with the host via a 64-bit, 66-MHz PCI bus, through an internal RAM. The high-frame-rate camera was a CCD device (model num-

ber CA-D6-256W, DALSA). Its resolution was 256×260 , and the frame rate was 955 fps. Images were transferred from the camera to the system in real-time. The host PC was a Dell Precision 670 with an Intel Xeon 2.80-GHz main CPU. This host CPU controlled the data transfer between PCI devices. In addition, it carried out pre-processing for simple image conversion, area cutout, and noise removal and performed numerical calculations using image features extracted by the co-processor.

4. MOMENT EXTRACTION OF NUMEROUS OBJECTS ON A PARALLEL ARRAY

This section describes the features that make it possible to extract the moments of over 1,000 objects at 1 kfps: pixel-parallel operation and object-parallel operation.

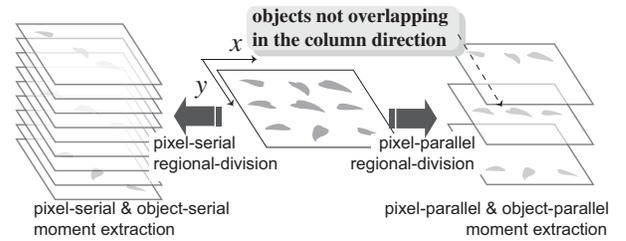


Fig. 2. Comparison of moment extraction approach.

The calculation of the moments consists of multiplication in each pixel and summation of all pixel data, as shown in Eq.(1). Therefore, pixel-parallel operation allows one moment extraction to be sped up because the multiplications are done simultaneously and the summation is done in parallel in rows/columns. We used a method based on 1-bit serial calculation [4], which did not require multipliers, to reduce the size of each element and increase the parallelism.

By incorporating this approach, our proposed object-parallel extraction solved the problem of increasing calculation time in proportion to the number of target regions [4]. This method allows the processing of multiple objects by using as many processing elements as possible at a time in the pixel-parallel structure. Fig.2 illustrates our method by comparing it with a typical calculation method that accesses data for one pixel at a time and performs moment extractions for multiple objects one by one.

In our method, for moments, the extraction in the pixel-parallel array is performed in column-parallel fashion. Therefore, we apply this object-parallel extraction by simultaneously processing multiple objects that do not overlap in the column direction. This allows us to decrease the total amount of calculation to about $O(\sqrt{n})$, where n is the number of objects.

The right side of Fig.4 shows the design of the circuit blocks for this operation. The function of each circuit block is described by Eq.(2).

$$m_{ij} = \sum_x x^i \left\{ \sum_y y^j I(x, y) \right\} = \sum_x x^i m_{0j}(x). \quad (2)$$

A column tree adder, in which elements are connected in a binary tree, calculates the moment for the y -coordinate m_{0j} . Each element consists of a 1-bit full adder and a register for the carry bit. Each block outputs a bit result every clock cycle, while selecting proper pixel data through the masking unit. The latency is $O(N)$ when the input data is 2^N . The x -coordinate is multiplied with each column result outside the array of column tree adders. For example, for the moments from 0th to 2nd order, m_{00} , m_{01} and m_{02} need to be calculated within the array. The moments m_{10} and m_{20} are calculated from m_{00} , and the moments m_{11} are in turn calculated from m_{10} .

The external processing unit consists of three blocks: a block for multiplying the x -coordinate, a block for separating the column results data for each region, and a block for adding all data. In actual implementation, the external processing unit contained two summation blocks to calculate the moments of multiple objects simultaneously.

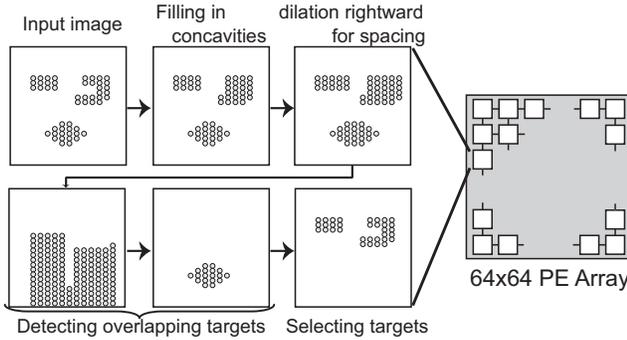


Fig. 3. Target selection.

On the other hand, this parallel extraction method requires pre-processing for selecting non-overlapping objects from any input pattern. We previously proposed a SIMD parallel algorithm based on a massively parallel architecture and inter-pixel asynchronous communication [4]. Fig.3 shows the flow of the algorithm.

The fundamental operation is the logical multiplication between the input image and the processed image shown in the bottom left of Fig.3, to which dilation in the column direction is applied. Then, we can obtain an image containing overlapping objects after applying a filling operation to the multiplied image. Using this image and the input image, an image containing non-overlapping objects is finally obtained. However, when the image includes objects with concavities in the column direction, the algorithm does not succeed. The filling-in of the concavities shown at the top center of Fig.3 avoids this problem. This algorithm can be described in a recursive fashion involving communication with only four neighbors to ensure convergence of the results after a certain number of iterations. This allows asynchronous communication between pixels and speeds up its execution. The algorithm was implemented as hardwired logic, and each processing element was connected to nearest neighbors in a two-dimensional 64×64 mesh array.

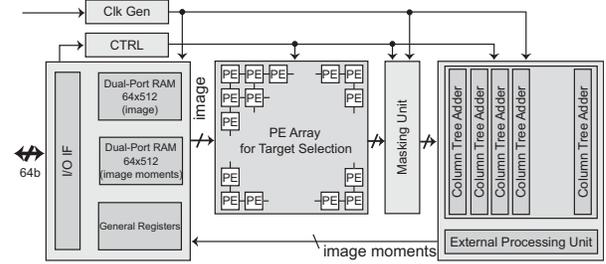


Fig. 4. Co-processor circuit block for moment extraction.

The total number of calculated objects was determined by the number of objects to be subjected to object-parallel extraction and the number of iterations. For example, when we extracted 8 objects simultaneously and repeated the extraction 8 times, the moments of 64 regions were obtained. These specifications were changed by instructions from the controller. The two cores implemented in FPGA (XC2V6000) each included the circuits shown in Fig.4. The selection circuit and the array of column tree adders used 73% of the available LUTs and 45% of the available FFs. The external logic for moment extraction used 5% of the LUTs and 6% of the FFs.

5. SYSTEM PERFORMANCE

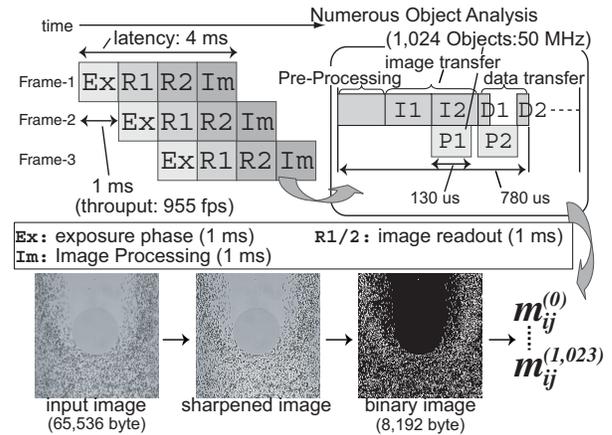


Fig. 5. Timing chart.

The overall timing chart is shown in Fig.5. The flow of operations in this chart was as follows: First, an image was captured (Ex) and stored in main memory through the capture board (R1/R2). Next, the image was sharpened and converted to a binary image on the main CPU (Pre-Processing) and transferred to the two co-processors (I1/I2). Each co-processor received a 256×128 image and extracted moments of numerous objects (P1/P2). Finally, the moments were transferred from the co-processors and stored in the main memory (D1/D2). Depending on the application, a recognition process using the extracted features may be added to the CPU.

In the co-processor core for moment analysis, 64×64 pixel divided image blocks in the 256×256 pixel input image were processed sequentially. The circuits were controlled

by a single clock at 50 MHz. Under these conditions, the moments from 0th to 2nd order of 1,024 regions (the unit extracted 8×8 regions from each block) were extracted in about $130 \mu s$, and the total time for image processing was $780 \mu s$. In addition, the number of dimensions and objects could be changed in the same circuits. For example, when we obtained 0th and 1st moments of 1,600 objects, it took about $120 \mu s$. Meanwhile, this operation has a possibility that some objects straddling both blocks were divided and each moment was obtained separately. After storing the moments, we corrected them in the CPU by determining corresponding pairs of moment values from the distance between them on the image and adding them.

As shown in Fig.5, the achieved performance was a throughput of 955 fps and a latency of 4 ms. This performance is considered sufficient to develop various real-time applications.

6. EXPERIMENTAL RESULTS

Here, we show results of two experiments using the developed system. The first experiment was pattern analysis using an artificial image, where we show the region information obtained from 0th, 1st, and 2nd moments. In the second experiment, we present the results of actual real-time image measurement at 955 fps using 0th and 1st moments. Each application used the co-processor setup described above.

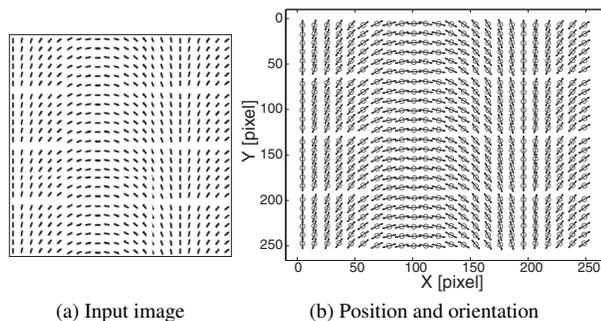


Fig. 6. Input image and results of pattern analysis.

Fig.6 shows the results of analysis of a test pattern including 576 objects. From the moments, we obtained the size, centroid, orientation, and aspect ratio of each object. Fig.6 graphically shows the results for the centroid and orientation.

The second experiment was real-time fluid measurement. A flowing fluid can be analyzed from the motion of over 1,000 particles in space based on the technique called Particle Image Velocimetry. This technique, however, is usually performed off-line due to the difficulty in realizing real-time operation.

In this experiment, we used expanded poly-styrene beads with a diameter of 1 mm. The measured fluid was generated by sucking air from one side of a space between parallel flat panels. A columnar obstacle was set near the center of the space. Fig.7(a) shows the measured points of the particles within four consecutive frames, and Fig.7(b) shows their velocity vectors. We can see that such a complex phenomenon was measured by moment-analysis of numerous objects.

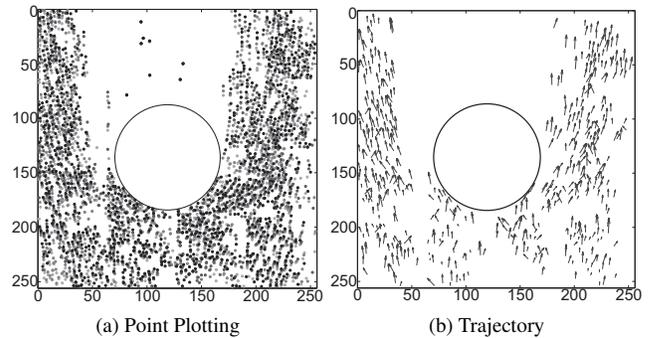


Fig. 7. Real-time fluid observation.

We also demonstrated high-speed shape measurement for a moving object using a pattern projection method based on the system described in [5]. In addition, the moments provide not only geometric features used in the above experiments, but also invariant features. By incorporating these features, our system will be effective in various real-time applications, such as flow control, particle manipulation, image classification in industrial inspection, active shape measurement for robotics, and so on.

7. CONCLUSION

High-speed image processing is highly promising for real-time feedback applications in automation, which require high-frame-rate observation and rapid acquisition of results. In order to develop new applications in such fields, we have developed a high-speed vision system for moment-based analysis of numerous objects. The distinctive features of the system are the provision of a dedicated co-processor and a novel calculation method for high-speed moment extraction. The system demonstrates the feasibility of image processing with high throughput. We also present experimental results showing the effectiveness of our system and its suitability for advanced application development in a wide range of fields.

8. REFERENCES

- [1] R. Mukundan and K. R. Ramakrishnan, *Moment functions in image analysis (Theory and Applications)*, World Scientific, 1998.
- [2] Alireza Moini, *VISION CHIPS*, Kluwer Academic Publishers, 2000.
- [3] U. Muehlmann, M. Ribo, P. Lang, and A. Pinz, "A new high speed CMOS camera for real-time tracking applications," in *Proc. of 2004 IEEE International Conference on Robotics And Automation*, 2004, pp. 5195–5200.
- [4] Y. Watanabe, T. Komuro, S. Kagami, and M. Ishikawa, "Parallel extraction architecture for image moments of numerous objects," in *Proc. of IEEE 7th International Workshop on Computer Architecture for Machine Perception (CAMP'05)*, 2005, pp. 105–110.
- [5] Y. Watanabe, T. Komuro, and M. Ishikawa, "955-fps real-time shape measurement of a moving/deforming object using high-speed vision for numerous-point analysis," in *Proc. of 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3192–3197.