

A High-Speed Vision System with In-Pixel Programmable ADCs and PEs for Real-Time Visual Sensing

Shingo Kagami, Takashi Komuro, and Masatoshi Ishikawa
Department of Information Physics and Computing,
University of Tokyo

Email: {Shingo_Kagami,Takashi_Komuro,Masatoshi_Ishikawa}@ipc.i.u-tokyo.ac.jp

Abstract— This paper describes a real-time vision system of which the pixel contains a photo detector, an analog-to-digital converter (ADC), and a digital processing element. The pixel-parallel processing array combined with a dedicated real-time control structure enables real-time visual processing at a high frame rate. The in-pixel programmable ADC offers flexible imaging capability with enhanced dynamic range. The system implementation issues are also described.

I. INTRODUCTION

Real-time sensing plays crucial roles in many situations where physical interactions take place. Visual sensing, among others, has significant advantages in that it can obtain huge amount of valuable information through contactless measurement. Recent increase of the demands on visual measurement reveals that conventional image sensors do not always offer satisfactory performance in some regards.

Temporal resolution of visual sensing, given by the sensor's frame rate, is one of those that have been gaining recognition [1]. In particular, visual feedback control of mechanical systems inherently requires frame rates higher than those realized by conventional image sensors to maintain sufficient feedback rates, typically over 1 kHz.

It is obvious that real-time recognition of high frame rate visual information is achievable only when all of image sensing, image processing, and data transfer from sensors to processors are guaranteed to complete subject to required time constraints. Thus real-time vision systems must be fast enough with respect to all of these three, and at the same time, must be equipped with some mechanisms to guarantee the time constraints.

One ideal solution, which mainly removes the problem of data transfer, is to integrate an image sensor and an image processor on a chip by embedding a processing element (PE) that is directly connected to a photo detector (PD) in each pixel. This kind of device is called a vision chip, or a computational sensor [2]. It performs pixel-parallel processing over images immediately after they are captured without time-consuming and power-consuming image transfer.

In this paper, we describe a high-speed vision system for real-time applications based on this "vision chip" approach. Its main feature is that it employs a programmable digital circuit for the PE, while many other vision chips

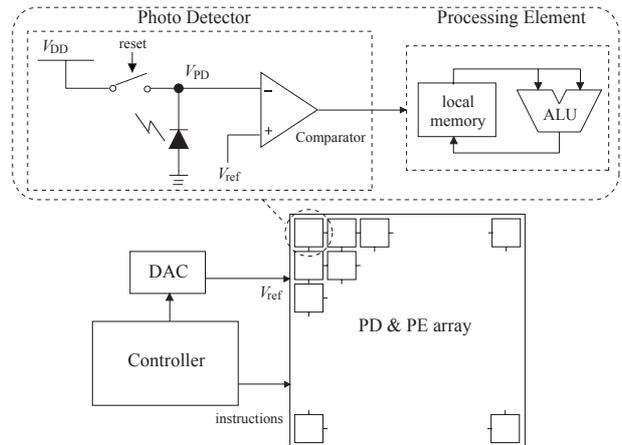


Fig. 1. Structure of the whole system.

employ analog circuitry with fixed functions. The PE combined with the readout circuit of the PD also acts as an analog-to-digital converter (ADC), which enables software-controlled conversion from incident light to digital pixel values. This PD/PE array combined with a hard real-time control structure offers powerful and flexible image sensing and image processing capabilities, and achieves visual processing at a frame rate over 1000 fps with enhanced dynamic range imaging.

We have already reported the PE array architecture [3] and the basic architecture for its real-time control [4]. This paper thereby focuses on the system implementation, and application of the programmable in-pixel ADC.

II. ARCHITECTURE

A. PE Array Architecture

The whole structure of the vision system is shown in Figure 1. It consists of a PD/PE array, and a controller that provides the array with digital and analog control signals.

Figure 2 shows the structure of the PE in each pixel [3]. All the pixels are controlled simultaneously in an SIMD (Single Instruction stream, Multiple Data stream) manner. The PE in each pixel has a bit-serial ALU, composed of a full adder circuit combined with some input/output multiplexers and a carry register, and a bit-wise random-access local memory. The ALU accepts two single-bit

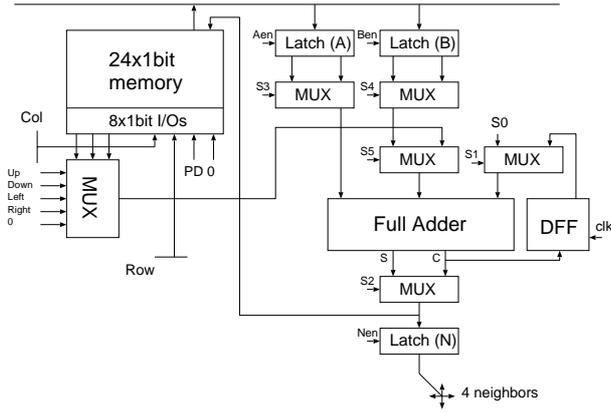


Fig. 2. Block diagram of the processing element.

operands in D-latches (Latches A and B in Fig. 2), which have been fetched from the local memory, and the result of a single-bit logical or arithmetic operation is written back to the local memory. Thus various kinds of image processing are executed by repeating these single-bit operations.

Each PE can communicate with its four neighboring PEs via their output ports consisting of D-latches (Latch N in Fig. 2). Inter-pixel operations needed in some spatial image processing such as edge detection and smoothing are thereby possible. In addition, adjacent multiple PEs can be joined and treated as a single combinational logic circuit by keeping the output latches transparent.

This PE-joining feature enables the joined PE block to execute global cumulative operations such as global summation and global-OR, or multi-bit ripple carry operations with each bit assigned to a separate PE. These operations can be used to implement global feature extraction or more advanced kinds of image processing, for example, parallel block matching or coordination transformation, which could not be implemented efficiently in plain pixel-parallel architectures.

B. Control Structure

In general, a SIMD PE array must be combined with an efficient control structure so that its potential is fully exploited. We have a distinct demand for our system in addition to merely maintaining instruction supply throughput: We have to incorporate a mechanism to guarantee real-time operations in order to support high frame rate visual processing, and moreover, to support real-time control of the PD and ADC with much higher temporal resolution.

We have already proposed a new microcontroller architecture [4] to meet this demand, in which a dedicated pipeline to control the PD/PE array (SIMD pipeline) is integrated with a RISC-type microprocessor. The basic configuration of the two pipelines is shown in Fig. 3. This architecture leaves the SIMD pipeline details, except the operation timing of each pipeline stage, to be freely designed for individual PE array architectures. Following the timing specification assures that any combinational use of the two pipelines never causes dynamic pipeline stalls, and thus real-time operations are guaranteed while

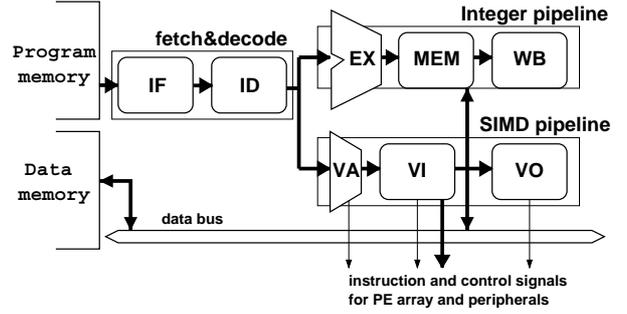


Fig. 3. Real-time control architecture for SIMD arrays.

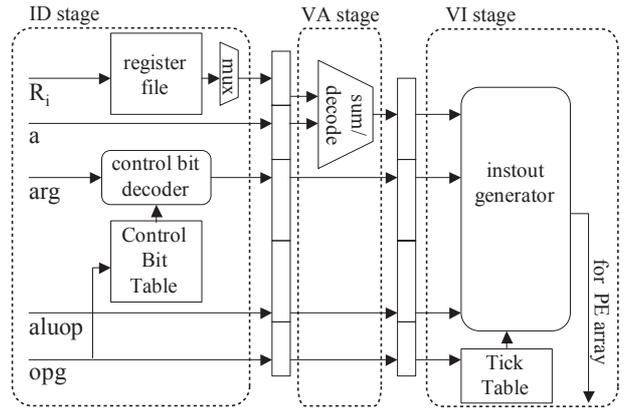


Fig. 4. Architecture of the SIMD pipeline.

maintaining the throughput.

Figure 4 shows the structure of the SIMD pipeline optimized for our PE array architecture stated above. A fetched instruction word is decoded in ID stage, and an effective address (used in the PE local memory) is generated in VA stage. After that, the expanded control signals are delivered to the PE array at the eightfold clock rate in VI stage. The instruction set and the control signal transition timing are fine-tunable to some extent by rewriting the tables embedded in the pipeline. The instruction set currently used is shown in Table I.

TABLE I
SIMD INSTRUCTION SET.

Instruction	Description
read(addr, f_{aen} , f_{ben})	Read data from the local memory and put them into the latch A and/or B
opw(op, addr, f_{clka})	Select operation, write the result to the local memory, and update the carry reg. if needed
opn(op, f_{clka})	Select operation, put the result into the latch N, and update the carry reg. if needed
nset(f_{nen})	Enable/disable the latch N
pdset(f_{pdr})	Reset/unreset PD
pulse(f_{clkb} , f_{clkc})	Give a pulse to the clock lines

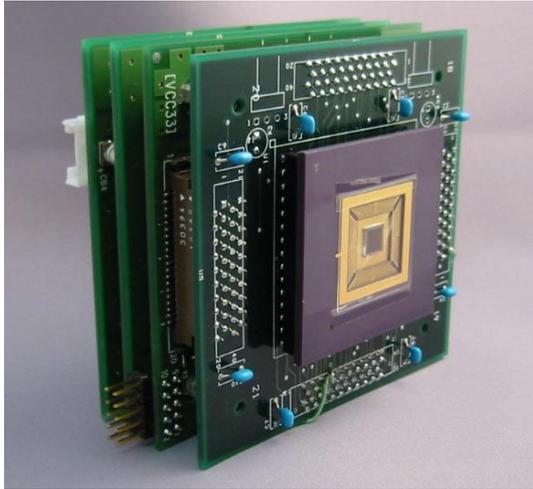


Fig. 5. Photograph of the implemented system, VCS-IV.

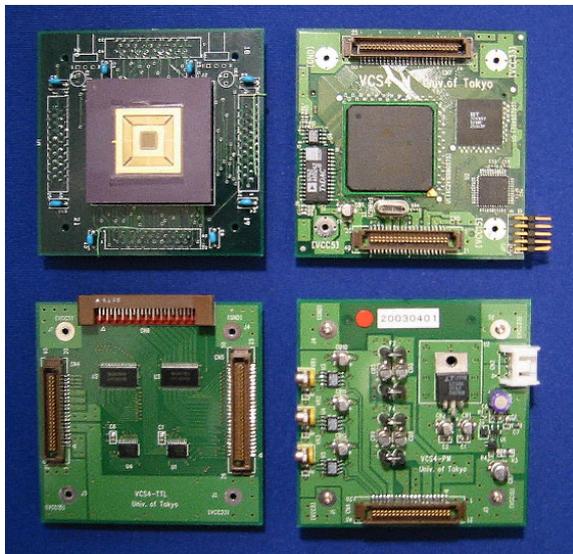


Fig. 6. Photograph of the system boards.

III. IMPLEMENTATION

In planning the implementation, we set two goals: extensibility as a research testbed, and usability in practical applications. These two goals often conflict with each other especially in deciding its form factor.

Figure 5 show a photograph of the implemented vision system, VCS-IV. The system consists of stackable circuit boards shown in Fig. 6. Dividing the system components into similarly-sized boards allows users to select what they need. Users can even develop a new additional board with their desired function because the number of stacks is structurally unlimited. Although the size of the board, $76 \times 76 \text{ mm}^2$, might not be small enough for desktop or mobile applications, it will be acceptable in many of industrial or scientific applications.

The sensor board, on the top of the system shown in Fig. 5, is equipped with a digital vision chip, which is a CMOS VLSI implementation of the PD/PE array described

TABLE II
EXECUTION TIMES OF SAMPLE PROGRAMS

program	steps	(time)
dilation, erosion (binary)	20	(2.0 μs)
edge detection (binary)	31	(3.1 μs)
edge detection (6bit)	193	(19 μs)
smoothing (binary)	47	(4.7 μs)
smoothing (6bit)	154	(15 μs)
0th moment ^{*1} (binary input)	124	(12 μs)
1st moment ^{*1} (binary input)	226	(23 μs)
centroid detection ^{*1} (binary input)	577	(58 μs)

*1 For 64×64 pixels.

in Sec. II-A. A micrograph of this chip is shown in Fig. 7. It was fabricated using $0.35 \mu\text{m}$ three-layer-metal CMOS process and integrates 64×64 pixels in $5.4 \times 5.4 \text{ mm}^2$ area. Each pixel, including the PD and the PE, consists of about 400 transistors.

The controller board, the second one from the top, contains an FPGA where the controller described in Sec. II-B is implemented. Any peripheral parts on this board, such as a program memory chip, are connected to this FPGA, and then they interface with external systems through the FPGA. Thereby we can keep connectivity with various kinds of external systems by just re-implementing an interface module in the FPGA.

These two boards are the mandatory components of the vision system. Optional components such as DC power supply and I/O level conversion are implemented in separate circuit boards in order to prevent the size of the board from being larger. The I/O level conversion board currently available is for 5V TTL interfaces, and used to connect the system to PCs.

The implemented system operated at the integer instruction rate of 10 MHz and the SIMD control signal rate of 80 MHz. Since the controller guarantees real-time instruction issuing with the temporal resolution of the instruction cycle time [4], the temporal resolution of real-time control is 100 ns. Execution times of several basic visual processing programs are shown in Table II.

Experimental results of several basic visual processing are shown in Fig. 8. A 6-bit grayscale image, its binarized and edge-detected image, and its centroid-detected image are shown in (a), (b), and (c) respectively.

IV. SOFTWARE A-D CONVERSION

To perform digital image processing at the pixel level, incident light illuminance must be converted to digital values at the pixel level. Since processing time and local memory capacity are limited, it is desirable to extract minimum necessary information in a flexible and efficient way during imaging, rather than achieve merely high-precision analog-to-digital (A-D) conversion.

The PD architecture of our system is shown at the top left of Fig. 1. It consists of a photodiode, a reset switch, and a comparator. First, the photodiode voltage

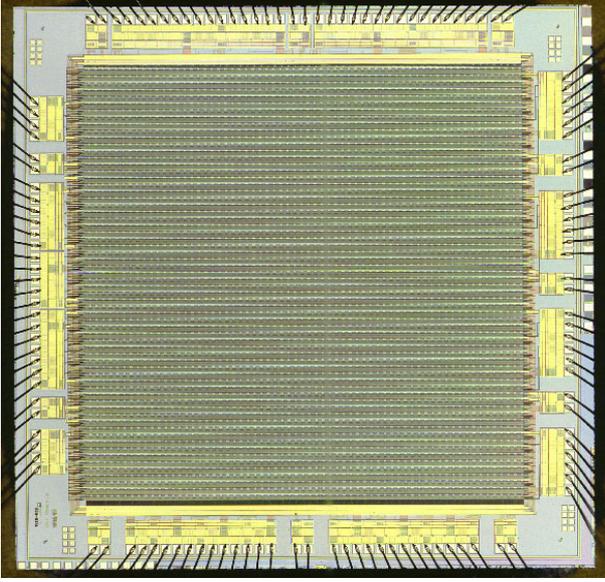


Fig. 7. Micrograph of the digital vision chip.

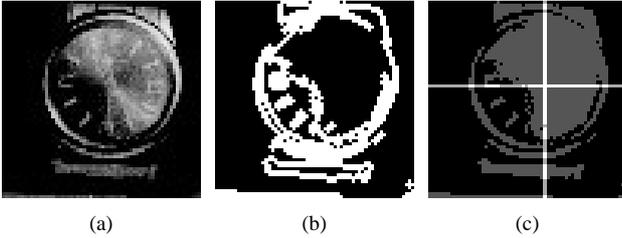


Fig. 8. Experimental results of basic visual processing. (a) 6-bit grayscale image (b) Edge detection (c) Centroid detection

V_{PD} is reset to V_{DD} , then left to float. Next, V_{PD} starts to be reduced, discharged by photocurrent. The comparator output holds logical 0 until V_{PD} drops under the reference input V_{ref} , and then it turns to logical 1. The brighter the incident light is, the earlier the comparator inversion occurs. Since the comparator output is always available for the PE as an input, a software counter associated with this PD input composes a software-controlled ADC. By controlling the PD readout timings and the reference voltage V_{ref} dynamically, which is possible because the controller guarantees fine-grained real-time control, various A-D conversion characteristics are obtained.

A. Predefined Scales

When desired quantization boundaries of photocurrent are given, the optimal V_{ref} curve and PD readout timings are uniquely determined [5].

Figures 9 (a) and (b) show images of a scene obtained with different A-D conversion scales, namely, a linear scale and a logarithmically compressing scale. The control schedules, V_{ref} curves and readout timings, are shown in Fig. 10, and the achieved A-D conversion scales are shown in Fig. 11. The maximum photo-integration time for the both is 8 ms, and the number of grayscale levels is 64 (i.e. 6-bit grayscale). We can see, by contrasting the image (b) with (a), that the dynamic range can be programmably



(a) linear (b) logarithmic (c) histogram-equalized

Fig. 9. Images obtained with different characteristics.

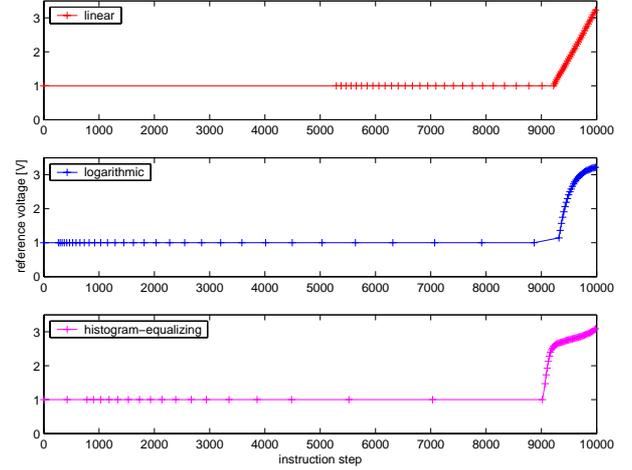


Fig. 10. A-D conversion schedules for several characteristics. One instruction step is treated as 1/10000 of the maximum photo-integration time.

enhanced.

B. Dynamic Adaptation

In a situation where lighting condition is unknown or varying, it is necessary for a sensor to be adaptable to the environment. As an example of dynamic adaptation, we describe a method of histogram-equalizing imaging.

This method determines A-D conversion boundaries of photocurrent $\{i_k\} (i = 0, 1, \dots, n)$ using frame-by-frame adaptation. Here, i_k is defined so that digital numbers $(n - k)$ are to be assigned to the analog quantities of the photocurrent i if $i_k \leq i < i_{k-1}$. It should be noted that i_1 is the brightest nontrivial boundary and i_{n-1} is the darkest (i_0 and i_n are fixed to ∞ and 0, respectively).

The A-D conversion boundaries for the next frame $\{i_k^{next}\}$ are computed using the boundaries $\{i_k\}$ and the cumulative histogram $\{H_k\}$ at the current frame as

$$i_k^{next} = i_{n-l+1} + \frac{H_n(n-k)/n - H_{l-1}}{H_l - H_{l-1}}(i_{n-l} - i_{n-l+1})$$

where l is the minimal natural number such that $H_l \geq H_n(n-k)/n$. H_k is defined as $H_k = \sum_{j=0}^{k-1} h_j$ where h_j is the number of the pixels of which the pixel values are j . This is a frame-by-frame binary search algorithm to find the desired cumulative histogram $H_n(n-k)/n$.

Once $\{i_k\}$ is given, the control schedule is generated just in a similar way to the previous cases. Figure 9 (c) shows an image of the same static scene as the previous examples, to which this histogram-equalizing algorithm

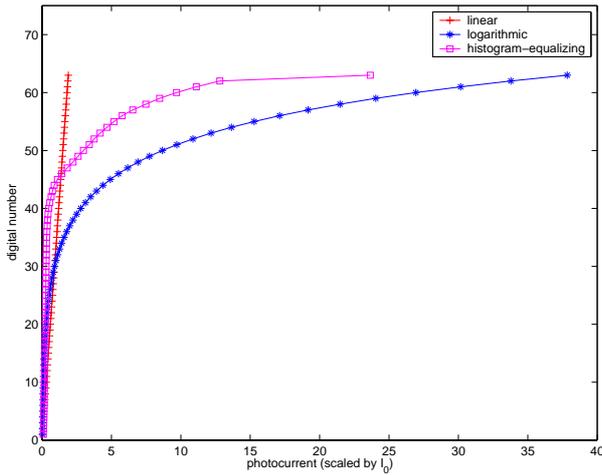


Fig. 11. Achieved A-D Conversion characteristics.

was applied. It took a few frames for the adaptation to converge. Figures 10 and 11 also show the schedule and the conversion scale at which this histogram-equalizing imaging converged.

It was also observed that the obtained scale slightly fluctuated. Discussion on stability when applied to dynamically changing scenes calls for further investigation.

C. Varying the Frame Rate

The results stated above are obtained with the frame rate fixed at a constant. On the other hand, we can control the frame rate dynamically since the pixel operations are fully programmable. This section demonstrates fixed A-D conversion scale imaging at different frame rates.

The conversion scale was fixed to a linear one, and the maximum photo-integration time was set to 1, 2, 4, 8, 16, and 32 ms. Figure 12 shows the obtained images. It is observed that the images share the same conversion scale, but have different noise levels. Clearly, the image with longer integration time has lower noise. It should be noted that, in our method, making the maximum exposure time longer never causes saturation, since it utilizes the time domain information as well as the voltage domain.

These results reveal the trade-off between the frame rate and the image noise. Software-controlled imaging allows users to select appropriate imaging characteristics based on their objectives or environments.

V. CONCLUSION

In this paper, a real-time vision system with programmably enhanced imaging capability for high frame rate applications has been described. Development of additional stackable boards, for example an MPU board with real-time operating system installed, is being planned.

REFERENCES

- [1] M. Ishikawa and T. Komuro, "Digital vision chips and high-speed vision systems," in *Digest of Technical Papers of 2001 Symposium on VLSI Circuits*, 2001, pp. 1–4.
- [2] A. Moini, *Vision Chips*. Norwell: Kluwer Academic Publishers, 2000.

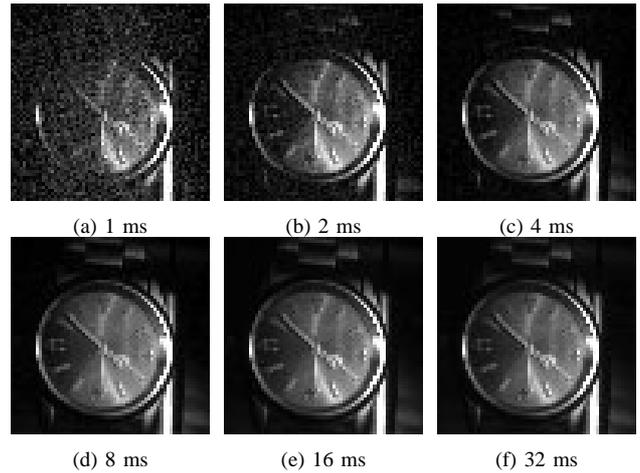


Fig. 12. Experimental results of fixed-scale, different-frame-rate imaging.

- [3] T. Komuro, S. Kagami, and M. Ishikawa, "A new architecture of programmable digital vision chip," in *Digest of Technical Papers of 2002 Symposium on VLSI Circuits*, 2002, pp. 266–269.
- [4] S. Kagami, T. Komuro, I. Ishii, and M. Ishikawa, "A real-time visual processing system using a general-purpose vision chip," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, 2002, pp. 1229–1234.
- [5] S. Kagami, T. Komuro, and M. Ishikawa, "A software-controlled pixel-level A-D conversion method for digital vision chips," in *2003 IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, 2003.