PAPER *Special Section on Novel Device Architectures and System Integration Technologies*

Vision Chip Architecture for Detecting Line of Sight Including Saccade

Junichi AKITA[†a)], ***Member***, **Hiroaki TAKAGI**[†], ***Nonmember***, **Takeshi NAGASAKI**[††], **Masashi TODA**[††], **Toshio KAWASHIMA**[††], ***and*** **Akio KITAGAWA**[††], ***Members***

SUMMARY Rapid eye motion, or so called saccade, is a very quick eye motion which always occurs regardless of our intention. Although the line of sight (LOS) with saccade tracking is expected to be used for a new type of computer-human interface, it is impossible to track it using the conventional video camera, because of its speed which is often up to 600 degrees per second. Vision Chip is an intelligent image sensor which has the photo receptor and the image processing circuitry on a single chip, which can process the acquired image information by keeping its spatial parallelism. It has also the ability of implementing the very compact integrated vision system. In this paper, we describe the vision chip architecture which has the capability of detecting the line of sight from infrared eye image, with the processing speed supporting the saccade tracking. The vision chip described here has the pixel parallel processing architecture, with the node automata for each pixel as image processing. The acquired image is digitized to two flags indicating the Purkinje's image and the pupil by comparators at first. The digitized images are then shrunk, followed by several steps of expanding by node automata located at each pixel. The shrinking process is kept executed until all the pixels disappear, and the pixel disappearing at last indicates the center of the Purkinje's image and the pupil. This disappearing step is detected by the projection circuitry in pixel circuit for fast operation, and the coordinates of the center of the Purkinje's image and the pupil are generated by the simple encoders. We describe the whole architecture of this vision chip, as well as the pixel architecture. We also describe the evaluation of proposed algorithm with numerical simulation, as well as processing speed using FPGA, and improvement in resolution using column parallel architecture.

key words: *vision chip, line of sight, saccade, pixel parallel processing, automata*

1. Introduction

Our eyes' line of sight (LOS), the direction where we are looking at, is expected to be applied to various types of human-computer interfaces. We also have the rapid eye motion, or the saccade, which is a very quick eye motion which always occurs regardless of our intention. Although the saccade is expected to be used for a new type of human-computer interface [1], [2], it is impossible to track it using the conventional video camera because of its speed which is often up to 600 degrees per second.

Some commercial systems for eye tracking including saccade exist, such as EyeLinkII [3], which consists of high speed camera and PC-based image processor, but their system is not compact for portable application. In addition, the processing delay, for example, up to 14 ms in EyeLinkII, may be a severe problem in real-time applications, since the image processing is performed in the pipeline manner to achieve high processing speed.

On the other hand, there are a lot of studies on 'Vision Chip,' which is an intelligent image sensor that has a photo receptor and processing circuitry on one chip [4]–[6]. Vision chip can process the acquired image information by keeping its spatial parallelism, and can achieve the very fast image processing, as well as compact image processing system as an integration of photo receptor and image processor. Generally, pixel parallel architecture results in low resolution because of pixel circuit size, such as 48×32 pixels in [6].

In this paper, we describe the vision chip architecture for calculating the line of sight from infrared eye image, with the capability of the saccade tracking using pixel parallel processing. The pixel circuit is designed for implementing the specified algorithm to calculate the line of sight to achieve high resolution. We describe the evaluation the operation speed by using FPGA. We also describe the column parallel architecture plan for higher resolution.

This vision chip is expected to implement the very compact line of sight detector compared with the conventional systems, and to expand the applications of the line of sight detector system.

2. Infrared Eye Image and Saccade

Infrared image of eye is suitable to find the line of sight. Figure 1 shows an example of infrared image of eye. Two

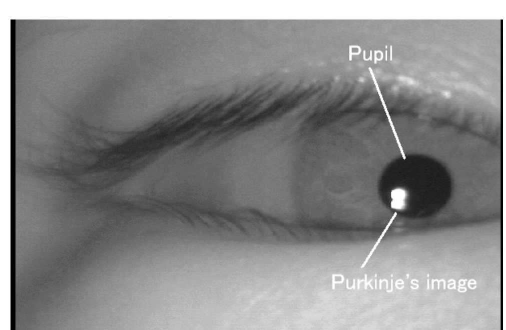


Fig. 1 Infrared image of eye.

Manuscript received April 10, 2006.
Manuscript revised June 16, 2006.

[†]The authors are with Kanazawa University, Kanazawa-shi, 920-1192 Japan.
[††]The authors are with Future University-Hakodate, Hakodate-shi, 041-8655 Japan.
a) E-mail: akita@is.t.kanazawa-u.ac.jp

DOI: 10.1093/ietele/e89–c.11.1605

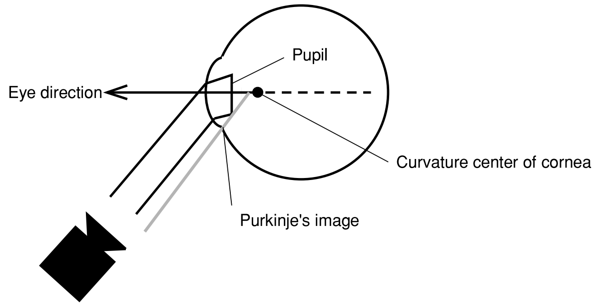


Fig. 2 Line of sight.

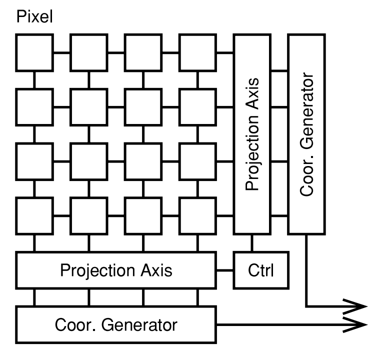


Fig. 3 Vision chip architecture.

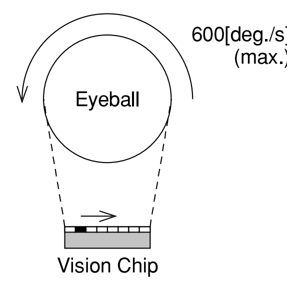


Fig. 4 Pixel speed of eye image on the vision chip.

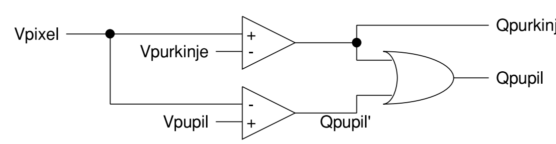


Fig. 5 Pupil and Purkinje's image detection circuit.

characteristic areas appear in the infrared image of eye. One is the pupil at the center of the iris, which appears as a black area, and the other is the Purkinje's image, which appears as a white spot around or inside the pupil.

The line of sight can be calculated by the relation of these two images' positions as shown in Fig. 2.

3. Fast Position Detection Architecture

The essential function of this vision chip is to extract the positions of the pupil and the Purkinje's image in order to determine and track the line of sight including the saccade. We applied the pixel parallel architecture for its implementation, where all the pixels have both photo receptor and processing circuitry. The pixel parallel architecture has a capability of very fast processing, while it often results in low resolution and low fill factor. We assumed the magnified eye image projected onto this vision chip for sufficient position resolution, with using fast tracking actuator to track the image of eye.

The conventional centroid detection performed in other vision chips employs the calculation of the momentum with object masking. The processing time of this algorithm depends on the number of objects, and it is difficult to guarantee the upper limit of operation time. We employed the centroid detection algorithm using shrinking processing performed by the node automata located at each pixel. This algorithm can guarantee the upper limit of processing time, since shrinking are performed in all the node automata in parallel manner.

The functional elements integrated on one pixel is composed of digitizer, node automaton for image shrinking and expanding, projection circuit onto x- and y- axes. There are also the priority encoders for coordinate generation, as well as transition control circuitry outside the pixel plain, as shown in Fig. 3.

The saccade is a very quick eye motion whose speed is up to 600 degrees per second. The time which the eye image across the vision chip for saccade, which is calculated as the time for the eye ball rotating 180 degrees, is 180/600=300 [ms]. Assuming the number of pixels on the vision chip as 100×100, the time for the eye image crossing one pixel is calculated as 300/100 = 3 [ms] as shown in Fig. 4. The required frame rate for processing to track saccade less than one pixel in the successive frame is estimated as 1/3 [ms] = 333 [fps].

Here, we describe the details of each element.

3.1 Digitizing

The digitizing procedure is implemented by the circuit shown in Fig. 5 for each pixel. The flag indicating the Purkinje's image, `Qpurkinje`, can be determined if the pixel is brighter than the reference, while the flag indicating the pupil, `Qpupil'`, can be determined if the pixel is darker than the references, V_{purkinje} and V_{pupil}, respectively.

The positions of these two images can be defined as the positions of their centroids. Since the Purkinje's image may exist inside the pupil, the image of the pupil can become the black circle with the small circle of the Purkinje's image dropped, as shown in Fig. 6(a). This shape of the pupil area may cause the calculation error of the centroid of the pupil, while the pupil area will be the certainly the circle in case of the Purkinje's image outside the pupil, as shown in Fig. 6(b), that makes the certain centroid of the pupil as the center of the pupil.

In order to avoid this problem, the certain flag indicating the pupil, `Qpupil`, should be generated as the logical OR

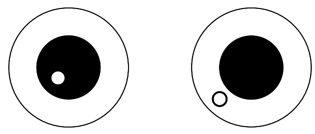


Fig. 6 Possible positions of pupil and Purkinje's image. (a)in case of Purkinje's image inside pupil, (b)in case of Purkinje's image outside pupil.

Fig. 7 Digitized image using detector circuit in Fig. 5.

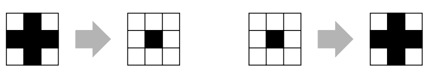


Fig. 8 Transition algorithms for the pupil area. (a) shrinking and (b) expanding.

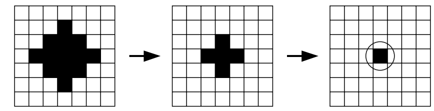


Fig. 9 The centroid detection by shrinking procedure.

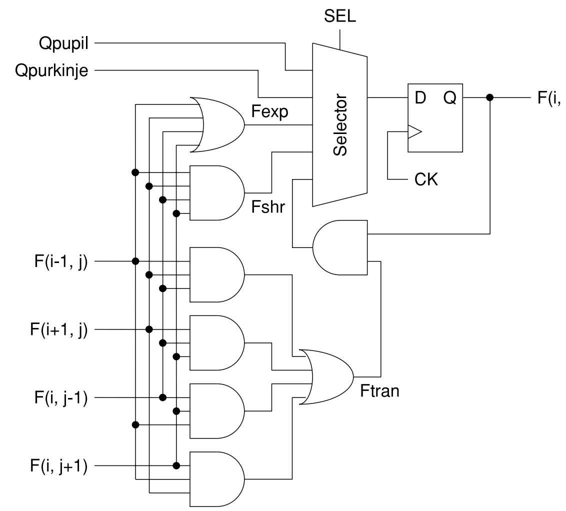


Fig. 10 Circuit of node automaton.

of the flag of the Purkinje's image, `Qpurkinje`, and flag indicating that brightness is darker than the reference of V_{pupil}, `Qpupil'`, as shown in Fig. 5. The small circle may appear if the Purkinje's image exists outside the pupil as shown in Fig. 6(b), but this small circle is smaller than the pupil area, and it disappears before the detection of the pupil in the following shrinking procedure described in Sect. 3.2.

Figure 7 shows the simulated digitized image for the case of the Purkinje's image inside the pupil. There are also other 'noise' areas such as the eyelashes, but they also disappear before the detection of the pupil as well. It is also found to be the 'white' gap around the Purkinje's image in Fig. 7 inside the pupil, which has the intermediate brightness between $V_{purkinje}$ and V_{pupil}. This gap can be eliminated as it expands, as described in Sect. 3.2.

3.2 Shrinking and Expanding

The image operations for the pupil and the Purkinje's image are carried out as the transition of node automaton placed in each pixel.

The shrinking procedure for the pupil area is implemented by the transitions of the automaton located at each pixel according to the transition rule in Fig. 8(a). The circle area of the flags indicating the pupil gets smaller and smaller in the procedure of shrinking transition, and finally disappears. The position of the smallest area just before it disappears can be regarded as the centroid of the circle area, as shown in Fig. 9.

The white gap can appear in the flags of the pupil in case of the Purkinje's image inside the pupil as shown in Fig. 7, which doesn't make the certain circle for the pupil area. In order to connect the areas of the Purkinje's image and the pupil in such case, some expanding transitions should be carried out prior to the shrinking procedure according to the rule in Fig. 8(b).

The global controller circuitry decides the transition rule for all automata for the pupil flags as follows.

1. a few steps of expanding transition
2. shrinking transition until all the area disappears

The position of the Purkinje's image is also calculated by the shrinking procedure similar to that for the pupil, while the expanding transition steps are not needed for the Purkinje's image flags, since they always form the circle or oval area in the pixel plain. The transition algorithm of the shrinking for the Purkinje's image is determined based on the simulations for eye images to perform the accurate centroid detection as follows.

- becomes '1,' if three of the four neighbouring automata have the value of '1'
- becomes '0,' otherwise

Figure 10 shows the designed circuit of node automaton to implement these operations. Flags indicating the pupil and the Purkinje's image, `Qpupil` and `Qpurkinje`, are given by the digitizer circuit shown in Fig. 5. The value of node automaton, `F(i,j)` is generated by the values of four neighbouring node automata, `F(i-1,j)`, `F(i+1,j)`, `F(i,j-1)` and `F(i,j+1)`, which is kept in D flip-flop, and changes according to the transition clock, `CK`. The values of expanding and shrinking transitions for the pupil, `Fexp` and `Fshr`, are generated by logical OR and logical AND of four neighbouring node automata's values, respectively. The value of shrinking transition for the Purkinje's image flags, `Ftran`, is also generated by the logical OR of four logical ANDs of three of neighbouring node automata's values. One of these values is selected and given to the D flip-flop by the select signal, `SEL`, which is controlled by the global controller circuitry as follows.

1. load the the pupil flag (D=`Qpupil`)
2. several expanding transitions (D=`Fexp`)
3. shrinking transitions until all disappear (D=`Fshr`)
4. load the the Purkinje's image flag (D=`Qpurkinje`)
5. shrinking transitions until all disappear (D=`Ftran`)

These procedures are evaluated by numerical simulations for eye image in Sect. 4.1.

3.3 Projection

The output of this vision chip is the coordinates of the pupil and the Purkinje's image in order to calculate the line of sight. Two steps of procedures are carried out in order to generate their coordinates for both the Purkinje's image and the pupil, respectively. The first is the projection of the automaton's values onto both *x*- and *y*- axes, and the second is coordinate generation of the projected images.

The values of the node automata, `F(i,j)` are projected onto the *x*- and *y*- axes, respectively during the transitions of node automata for both the Purkinje's image procedure and the pupil procedure, respectively, as shown in Fig. 11. The projection to *y*-axis at *j*-th row, `Pyj`, can be implemented by the chain of the logical ORs of the node automata values without clock signal for fast operation as shown in Fig. 12. The projection to *x*-axis at *i*-th column, `Pxi`, is also implemented as well.

Not only the pupil or the Purkinje's image but also other noise images may exist on the focal plain as well as their projected images onto both axes. The target image of each operation for the pupil and the Purkinje's image is the largest area, which disappears at last during the shrinking transitions.

The disappearing timing of both pupil and Purkinje's image during the shrinking transitions can be detected when all of the projected images onto the axes are going to disappear. This condition is described by the number of the projected values of '1' onto both axes, $N_b(x)$ and $N_b(y)$ is 2 or 1, since all flags can't disappear in the next shrinking transition if there are neighbouring flags of '1,' while all the flags disappear at the next transition if there are no pairs of neighbouring pixels of '1,' as shown in Fig. 13. The detection of $N_b(x)$ and $N_b(y)$ going to zero can be determined detecting there are not three neighbouring '1's of projected images. This condition can be detected by the logical ANDs of the neighbouring three projected images on both axes as shown in Fig. 14.

3.4 Coordinate Generation

The final procedure is the coordinate generation for the pupil and the Purkinje's image. The positions of the projected images at the end of transition, just before all disappear, detected by the circuitry in Fig. 14 are expected to indicate the centroid of the areas of the pupil and the Purkinje's image. Their positions can be decided by the priority encoders lo-

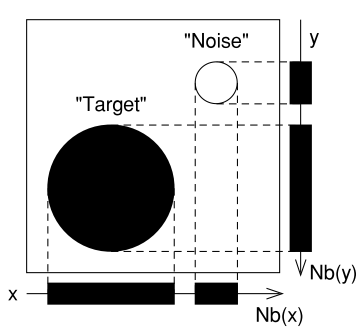


Fig. 11 Projected flags and the number of projected flags.

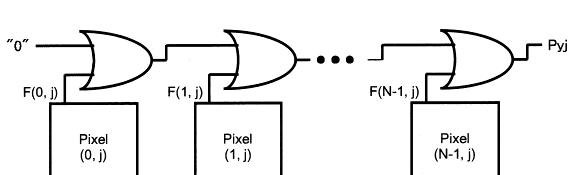


Fig. 12 Chain of logical ORs for generating projection.

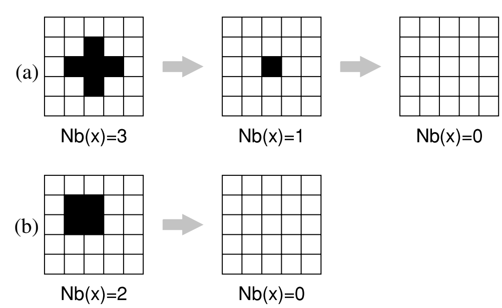


Fig. 13 The number of projected values of '1' when disappears during the shrinking transition for two cases.

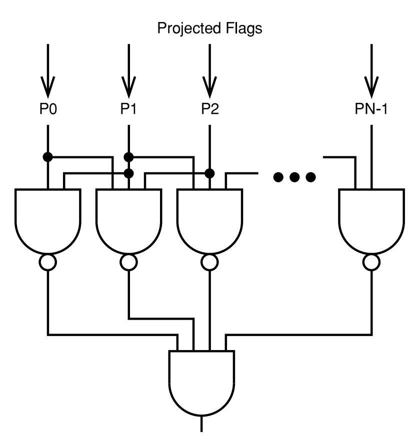


Fig. 14 Transition end detection circuitry.

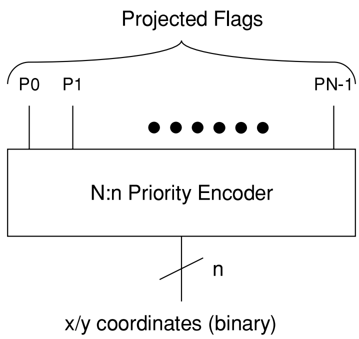


Fig. 15 Priority encoders for coordinate generation.

cated at x- and y- axes, respectively, as shown in Fig. 15, which generates the binary position (coordinate) of '1' in N inputs.

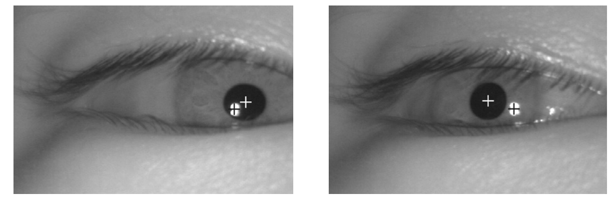
4. Evaluation

4.1 Operation of Automata

We carried out the numerical simulation to evaluate the shrinking procedure described in Sect. 3.2 for eye images. The infrared eye images are digitized by the adequate threshold, and the shrinking and expanding procedures are carried out in the numerical simulator based on the operations of the designed node automaton in Fig. 10. The positions of both the pupil and the Purkinje's image detected by these procedures are shown in Fig. 16 for the cases of the Purkinje's image both inside and outside the pupil. The positions of the pupil and the Purkinje's image are correctly detected.

4.2 Evaluation on FPGA

We carried out FPGA implementation in order to evaluate the processing speed of the proposed vision chip architecture. We implemented both 8×8 and 16×16 arrays of pixels of node automata as well as the registers to hold the digitized value, projection and coordinate generation circuitry on Altera EP1C20F324. The digitized values for the pupil and the Purkinje's image are externally fed into the registers of pixels, and all of the processing procedures are carried out. As a result, we obtained the correct result of generated coordinates. The critical path of whole pixel array circuitry is the chain of logical ORs for projection, since this chain exist across all pixels, while the automata has the connections only to the neighbouring automata, which doesn't increase even if pixels increase. The critical path delay for operation is determined by the length of the crossing path, or the number of pixels. The measured maximum delay of the circuit is 27.7 [ns] for 8×8 pixels, and 30.8 [ns] for 16×16 pixels. This increase of the delay is expected to be on the increased eight logical OR in the projection chain. The estimated maximum delay for $n \times n$ pixels in the projection chain, $T_d(n)$, is to be as follows.



(a) (b)

Fig. 16 Detected positions of pupil and Purkinje's image. White and black crosses indicate positions of pupil and Purkinje's image, respectively. (a) Purkinje's image inside pupil, (b) Purkinje's image outside pupil.

$$T_d(n) = 24.6 + 0.39 \times n \,[\text{ns}] \quad (1)$$

The maximum number of shrinking steps is determined by the half of the number of pixels, since one pixel at the edge of the pupil or the Purkinje's image areas disappear in each shrinking step. The total number of shrinking steps to detect the positions of both the pupil and the Purkinje's image is, therefore, equal to the number of pixels in one edge of pixel plain, n. The estimated maximum frame rate, F for 100×100 pixels can be estimated with $T_d(100) = 63.4$ [ns] as follows.

$$F = 1/(63.4\,[\text{ns}] \times 100) = 158\,[\text{kfps}] \quad (2)$$

This estimated frame rate derived from the operation time in coordinates generation is fast enough for saccade tracking. The actual frame rate is restricted by other factors, mainly the integration time of photo receptor, and the design for sensitive photo receptor will be required.

5. Column Parallel Architecture

The position detection algorithm described above implemented by node automata in pixel parallel manner is fast enough, more than 100 times faster than the required operation speed for saccade tracking.

One solution for achieving higher resolution is to employ the column parallel architecture instead of pixel parallel architecture. Column parallel architecture has the processing element in each column as shown in Fig. 17, and it sequentially performs the operations for each row. The total operation speed is slower than that of pixel parallel architecture, but the higher resolution and higher fill factor can be achieved since the pixel only has the photo receptor and read-out circuit.

The position detection algorithm using shrinking procedure can also be implemented in column parallel arcitecture with node automaton and flag memories as shown in Fig. 17.

For example, the total number of shrinking steps in VGA (640×480) resolution is estimated as about 300 ($480/2 = 240$ for the pupil and 60 for the Purkinje's image). The node automaton in each column makes 480 transitions to perform the shrinking operation for one column. To achieve the frame rate of 1 [kfps], the required operation frequency for node automaton in each column, $f_{\max}$ is estimated as follows.

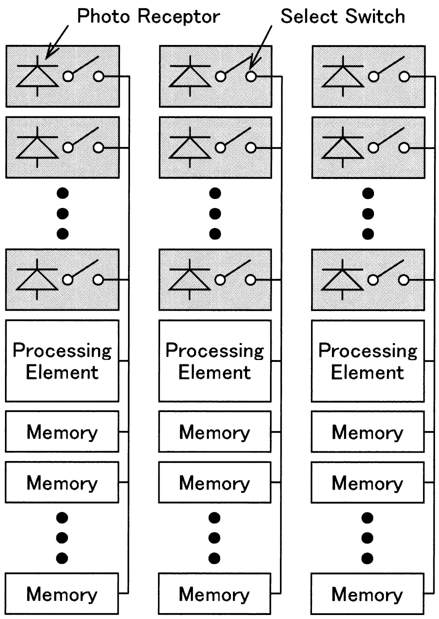


Fig. 17 Column parallel architecture for position detection.

Table 1 Estimated f_{max} of node automaton in column parallel architecture.

Resolution	Frame Rate [fps]	f_{max} [MHz]
VGA(640×480)	1k	144
VGA(640×480)	500	72
QVGA(320×240)	1k	36
QVGA(320×240)	500	18

$$f_{max} = 1\,[\text{kfps}] \times 480 \times 300 = 144\,[\text{MHz}] \tag{3}$$

This operation frequency can be achieved by conventional CMOS technologies. The estimated f_{max} for various resolutions and frame rates are summarized in Table 1.

The circuit implementation will be discussed and evaluated in our future work.

6. Conclusions

In this paper, we proposed the vision chip architecture for detecting line of sight including the saccade with pixel parallel processing. Its processing speed is estimated to be fast enough for saccade tracking. We also proposed the column parallel architecture for higher resolution. The detailed designs including sensitive photo receptor will be reported our future work.

This vision chip is expected to implement the saccade tracking system for various types of human-computer interfaces, as well as the very compact line of sight detector compared with the conventional systems, and to expand the applications of the line of sight detector system.

References

[1] J. Triesch, D.H. Ballard, M.M. Hayhoe, and B.T. Sullivan, "What you see is what you need," J. Vision, vol.3, no.1, pp.86–94, 2003.
[2] S.S. Intille, "Change blind information display for ubiquitous computing environments," Proc. UbiComp2002, pp.91–106, 2002.
[3] http://www.monte.co.jp/srr01.htm
[4] T. Komuro, S. Kagami, and M. Ishikawa, "A dynamically reconfigurable simd processor for a vision chip," IEEE J. Solid-State Circuits, vol.39, no.1, pp.265–268, 2004.
[5] A. Åström, J.E. Eklund, and R. Forchheimer, "Global feature extraction operations for near-sensor image processing," IEEE Trans. Image Process., vol.5, no.1, pp.102–110, 1996.
[6] T. Komuro, I. Ishii, M. Ishikawa, and A. Yoshida, "A digital vision chip specialized for high-speed target tracking," IEEE Trans. Electron Devices, vol.50, no.1, pp.191–199, 2003.

Junichi Akita was born in Nagoya, Japan in 1970. He received B.S., M.S. and Ph.D. degrees in electronics engineering from the University of Tokyo, Japan in 1993, 1995 and 1998 respectively. He joined the Department of Computer and Electrical Engineering, Kanazawa University as a research associate in 1998. He moved to the Department of Media Architecture, Future University — Hakodate as an assistant professor in 2000. He moved to the Department of Information and Systems Engineering, Kanazawa University as an assistant professor in 2004. His main research interest is in analog parallel signal processing VLSI architecture and its applications. He is also interested in electronics systems including VLSI systems in the applications of human-machine interaction and human interface. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan, Information Processing Society of Japan, and the Institute of Image Information and Television Engineering.

Hiroaki Takagi was born in Toyama, Japan in 1982. He received B.S. degree from the Kanazawa University, Japan in 2005. He is currently a student in the master course at Kanazawa University.

Takeshi Nagasaki was born in Hokkaido, Japan in 1969. He received B.E., M.E., and Ph.D. degrees from Hokkaido University in 1992, 1994 and 1998, respectively. From 1997 to 2000, he worked on B.U.G. Inc. Since 2000 he has been with Future University — Hakodate, where he is currently an assistant professor of Media Architecture. His current research interests are computer vision and wearable system. He is member of Institute of Electronics, Information and Communication Engineers of Japan, Information Processing Society of Japan, and the Institute of Electrical and Electronic Engineers.

Masashi Toda was born in Hamamatsu, Japan in 1969. He received B.S. from the University of Tokyo, Japan in 1993, and M.S. and Ph.D. in electro-informatics engineering from Hokkaido University, Japan in 1995 and 1998, respectively. From 1998 to 2001, he was a researcher in IS Labo., SECOM Co., Ltd.,Japan. From 2001 to 2005, he was an assistant professor in School of Systems Information Sciences in Future University-Hakodate, Japan. Since 2005, he has been an associate professor in Future University-Hakodate. His main research interest is in sensing architecture. He is also interested in image processing technology, wearable computing, ubiquitous computing, and educational information system. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan, and Information Processing Society of Japan.

Toshio Kawashima was born in Hokkaido, Japan in 1957. He received B.E., M.E., and Ph.D. degrees from Hokkaido University in 1980, 1982 and 1990, respectively. From 1984 to 2000, he worked at the Department of Information Engineering of Hokkaido University, where he was an associate professor. In 2000 he moved to Future University Hakodate as a professor of the Department of Media Architecture. He is engaged in research on multi-sensor system for human assistance and digital archives for historical documents. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan, and Information Processing Society of Japan, Association for Computing Machinery, and the Institute of Electrical and Electronic Engineers.

Akio Kitagawa was born in Shiga, Japan in 1961. He received B.S. and M.S. degrees from Nagoya Institute of Technology in 1985 and 1987, respectively. From 1989, he worked at the Faculty of Engineering, Kanazawa University as a research associate. From 1995, he has been an associate professor at Kanazawa University. From 1999 to 2001, he was also an assistant professor at VLSI Design and Education Center, University of Tokyo. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan, and Information Processing Society of Japan, and the Institute of Electrical and Electronic Engineers.