P23

Vision Chip Architecture for Saccade Tracking

Junichi Akita* Hiroaki Takagi* Takeshi Nagasaki** Masashi Toda**
Toshio Kawashima** Akio Kitagawa*

*Kanazawa University
Kakuma, Kanazawa, 920-1192 Japan
Phone: +81-76-234-4864 / FAX: +81-76-264-6404 / akita@is.t.kanazawa-u.ac.jp
**Future University-Hakodate

1 Introduction

Rapid eye motion, or the saccade, is a very quick eye motion which always occurs regardless of our intention. Although the saccade is expected to be used for new type of computer-human interface[1, 2], and 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 a photo receptor and processing circuitry on one chip, which can process the acquired image information with keeping its spatial parallelism[4, 5].

In this paper, we describe the vision chip architecture that has the capability of the saccade tracking with pixel parallel processing.

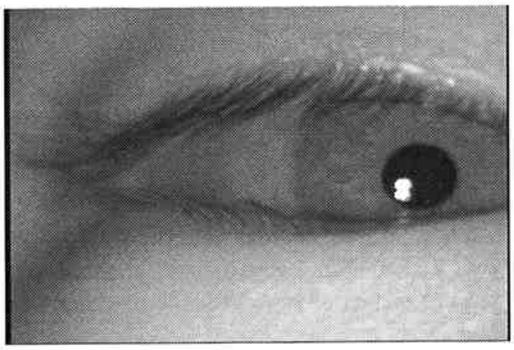

Figure 1: Infrared image of eye.

2 Infrared Eye Image

Infrared image of eye is suitable to find the direction of eye. Figure 1 shows an example of infrared image of eye. Two characteristic areas appear in the infrared image of eye. One is the pupil at the center of the iris which appears as an black area, and the Purkinje's image which appears as a white spot around or inside the pupil. The direction of the eye can be determined by the positions of these two images as shown in Fig.2.

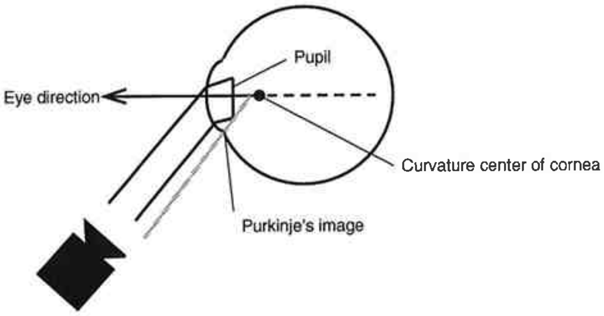

Figure 2: Direction of eye.

3 Fast Position Detection Architecture

The essential function of this vision chip is to extract the positions of both the pupil and the Purkinje's image in order to find and track the direction of the eye including the saccade. We assume the pixel parallel architecture for its implementation, where all the pixels have both the photo receptor and processing circuitry. The pixel parallel architecture has a capability of very fast processing instead of low resolution and low fill chip architecture. We implemented both the 8×8 and 16×16 arrays of pixels of node automata as well as the register 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 each pixel, and all the processing procedures are carried out, and we obtained the correct result of generated coordinates.

The critical path of whole pixel array circuitry is expected to be the chain of logical ORs for projection, since this chain exist across all the pixels, while the automata has the connections only to the neighbouring automata, which is not expected to increase for larger number of pixels. The measured maximum delay of the circuit is 27.7[ns] for 8×8 pixels, as well as 30.8[ns] for 16×16 pixels. The increase of the delay of 3.1ns is expected to be on the increased chain of eight logical ORs for projection. The estimated maximum delay for $n \times n$ pixels $T_d(n)$ is to be:

$$T_d(n) = 24.6 + 0.44 \times (n - 1) \quad [\mathrm{ns}] \tag{1}$$

The maximum number of transitions is also estimated to be n for the worst case.

The estimated maximum delay for 100×100 pixels is to be 68.2[ns], and the frame rate for this case is $1/68.2[\mathrm{ns}] \div 100 = 146[\mathrm{kfps}]$, which is fast enough for saccade detection.

5 Conclusions

We proposed the vision chip architecture that has a capability of eye tracking including the saccade by pixel parallel processing. Its processing speed is estimated to be enough for saccade tracking. The detailed designs including sensitive photo detector will be reported our future work.

References

[1] J.Triesch *et al.*, "What you see is what you need," *Journal of Vision*, No.3, pp.86–94, 2003.

[2] S.S.Intille, "Change Blind Information Display for Ubiquitous Computing Environments," *UbiComp2002*, LNCS 2498, pp.91–106, 2002.

[3] J.Akita *et al.*, "An image sensor with fast objects' position extraction function," *IEEE Transactions on Electron Devices*, Vol.50, No.1, pp.184-190, 2003.1.

[4] T.Komuro, S.Kagami, M.Ishikawa, "A Dynamically Reconfigurable SIMD Processor for a Vision Chip", *IEEE Journal of Solid-State Circuits*, Vol.39, No.1, pp.265–268, 2004.1.

[5] J.E.Eklued *et al.*, "Global Feature Extraction Operations for Near-Sensor Image Processing," *IEEE Trans. Image Processing*, Vol.5, No.1, pp.102–110, 1996.

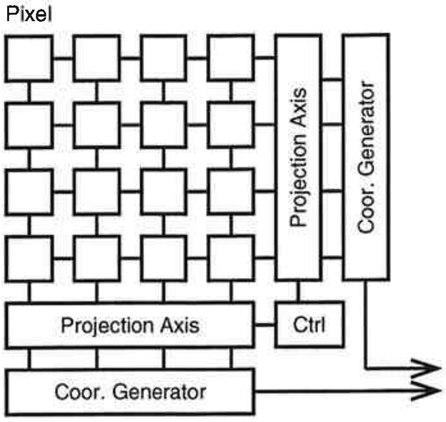

Figure 3: Vision chip architecture.

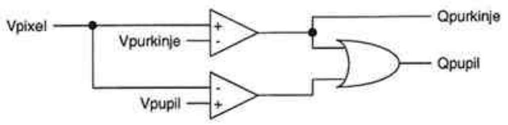

Figure 4: Pupil and Purkinje's image detection circuit.

factor. We assumed the magnified eye image projected onto this vision chip for sufficient position resolution with having fast tracking actuator to track the image of eye.

The function elements integrated on one pixel is composed of digitizer, node automata for image shrink and expand, projection onto $x-$ and $y-$ axes as well as coordinate generation circuitry on both axes outside the pixel plain, as shown in Fig.3. Here we describe the details of each function element.

3.1 Digitize

The digitize procedure is implemented by the circuit shown in Fig.4 for each pixel. The Purkinje's image can be determined if the pixel is brighter than the reference, while the pupil can be determined if the pixel is darker than the reference. The positions of these two images can be defined as the positions of their centroids. Since the Purkinje's image can be 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.5(a). This shape of the pupil may occur the error of the centroid of the pupil, while the pupil will become the certainly the circle in case of the Purkinje's image outside the pupil as shown in Fig.5(b) that makes the certain centroid of the pupil as the center of the pupil.

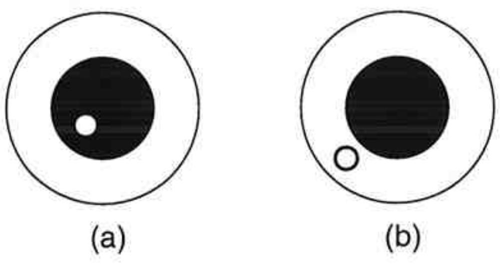

Figure 5: Possible positions of pupil and Purkinje's image. (a)in case of Purkinje's image inside pupil, (b)in case of Purkinje's image outside pupil.

Figure 6: Digitized image using detector circuit in Fig.4.

In order to avoid this problem, the flag indicating the pupil is determined by the logical OR of the Purkinje's image and the pixel whose brightness is darker than the reference as shown in Fig.4. The small circle may appear if the Purkinje's image exists outside the pupil as shown in Fig.5(a), but this small circle is smaller than the pupil area, and it disappears before the detection of the pupil in the following shrink procedure.

Fig.6 shows the simulated digitized image for the case of the Purkinje's image inside the pupil. There are also another 'noise' such as the eyelashes, but they also disappear before the detection of the pupil as well. It is also be found the 'white' gap around the Purkinje's image in Fig.6 inside the pupil, which has the brightness between the references of the pupil and the Purkinje's image. This gap can be eliminated the expansion transition as described the following procedure.

3.2 Shrink

The shrink procedure is implemented by the transitions of the automata located at all the pixels according to the transition rule in Fig.7(a). The circle becomes smaller and smaller in the procedure of shrink transition, and finally disappears. The position of the smallest area just before it disappears can be regarded as the centroid of the circle area, and its coordinate is generated in the following procedures. Each pixel has two automata for transitions of both the flags of the pupil and those of the Purkinje's image.

The white gap can be appear in the flags of the pupil in case of the Purkinje's image inside the pupil as shown in Fig.6, which doesn't make the certain cir-

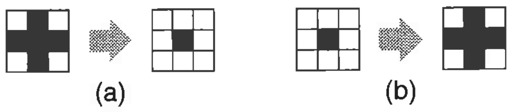

Figure 7: Transition algorithms. (a)shrink and (b)expansion.

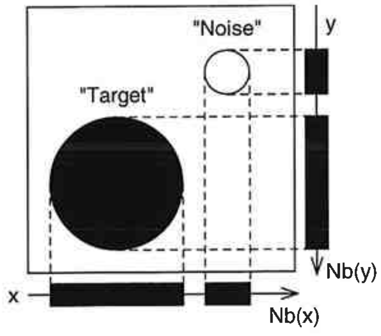

Figure 8: Projected flags and the number of projected flags.

cle for the pupil area. In order to connect the areas of the Purkinje's image and the pupil in such case, some expansion transitions should be executed prior to the shrink procedure by the rule in Fig.7(b).

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

1. a few steps of expansion transition
2. shrink transition until the area disappears

The expansion transition steps is not needed for the Purkinje's image flags, since always form the circle area in the pixel plain.

3.3 Projection

The output of this vision chip is the coordinates of the pupil and the Purkinje's image. Two steps of procedures are carried out in order to generate their coordinates. The first is the projection of the flags indicating the centroids onto both the $x-$ and $y-$ axes, and the second is coordinate generation of the projected images.

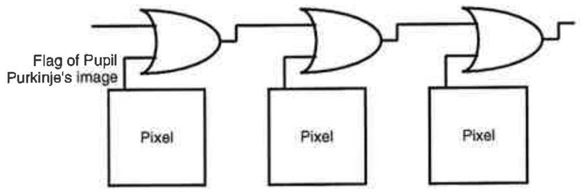

Figure 9: Chain of logical OR for generating projection.

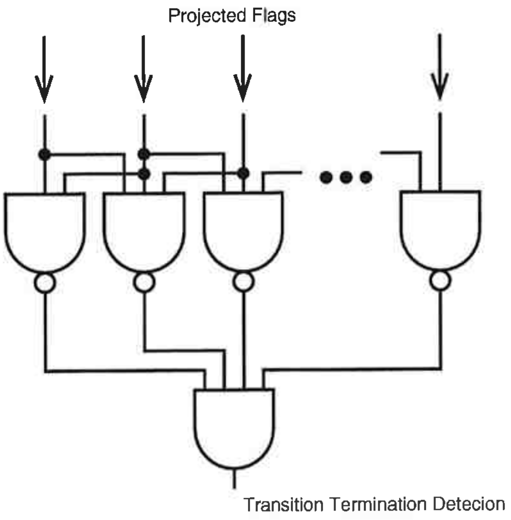

Figure 10: Transition end detection circuitry.

The flags indicating the centroids of both the pupil and the Purkinje's image are projected onto the $x-$ and $y-$ axes respectively during the transitions as shown in Fig.8. The projection can be implemented by the chain of the logical OR of the pixels as shown in Fig.9. Not only the pupil or the Purkinje's image but also other noise images exist on the focal plain as well as their projected images onto both axes. The target images are the largest in each flag plain so that they disappear at last during the shrink transitions. The disappearing timing of both the pupil and the Purkinje's image can be detected when all the projected images onto the axes are going to disappear, or the number of the projected pixels onto both the axes, $N_b(x)$ and $N_b(y)$ are going to zero. The detection of $N_b(x)$ and $N_b(y)$ going to zero can be detected by the logical AND of the neighbouring three pixels on both axes as shown in Fig.10, since the existence of the neighbouring pixels of '1' indicates all pixels will not disappear at next transition, while all the pixels disappear at the next transition if there are no pairs of neighbouring pixels of '1'.

3.4 Coordinate generation

The final procedure is the coordinate generation for the pupil and the Purkinje's image. The positions of the projected pixels at the end of transition detected by the circuitry in Fig.10 are expected to indicate the centroid of the areas of the pupil and the Purkinje's image, and their positions can be decided by the priority encoders which generates the binary position (coordinate) of '1' in n inputs.

4 Evaluation on FPGA

We carried out FPGA implementation in order to evaluate the processing speed of the proposed vision