

Column-Parallel Architecture for Line-of-Sight Detection Image Sensor Based on Centroid Calculation

Hayato Kawakami Satoru Igarashi Yuta Sasada Junichi Akita

School of Electrical and Comp. Eng., Kanazawa University Kakuma, Kanazawa, Ishikawa, Japan
TEL: +81-76-234-4864 E-mail: kawakami@ifdl.jp akita@is.t.kanazawa-u.ac.jp

1. Introduction

Line-of-sight (LoS) detection, or detecting where the user is looking at, is an attracting as a new type of user interfaces [1]. The eyeball also moves very rapidly, which is called as saccade, and it is also expected to be applied for another type of user interfaces [2], however, real-time tracking of saccade is difficult by using the conventional image processing systems for their long processing time and/or latency against the speed of saccade.

In this paper, we propose the column-parallel LoS detection architecture for the integrated image sensor, which can track the saccade.

2. LoS Calculation

The LoS can be calculated by the position of the pupil in the infrared image of the user's eye, which is observed as a black area [3], as shown in Fig.1. The position of the pupil, whose shape is a circle, can be defined as a centroid of the pupil (digitized black) area with the sub-pixel accuracy.

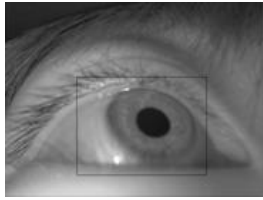


Fig.1: Example of an infrared image of eye

The centroid of the area, (\bar{x}, \bar{y}) with the binary flag of '1' for the pupil, can be calculated as follows.

$$\left(\bar{x} = \frac{\sum_x \sum_y x p_{xy}}{\sum_x \sum_y p_{xy}}, \quad \bar{y} = \frac{\sum_x \sum_y y p_{xy}}{\sum_x \sum_y p_{xy}} \right) \quad (1)$$

As shown in eq.(1), the values of $\sum_x \sum_y p_{xy}$, $\sum_x \sum_y x p_{xy}$ and $\sum_x \sum_y y p_{xy}$ are required in calculation of the centroid.

3. Column-parallel Architecture for Centroid Calculation

Fig.2 shows the designed column-parallel architecture for centroid calculation, to realize both high resolution, such as VGA or above, and high frame rate, such as 500[fps] of above [4].

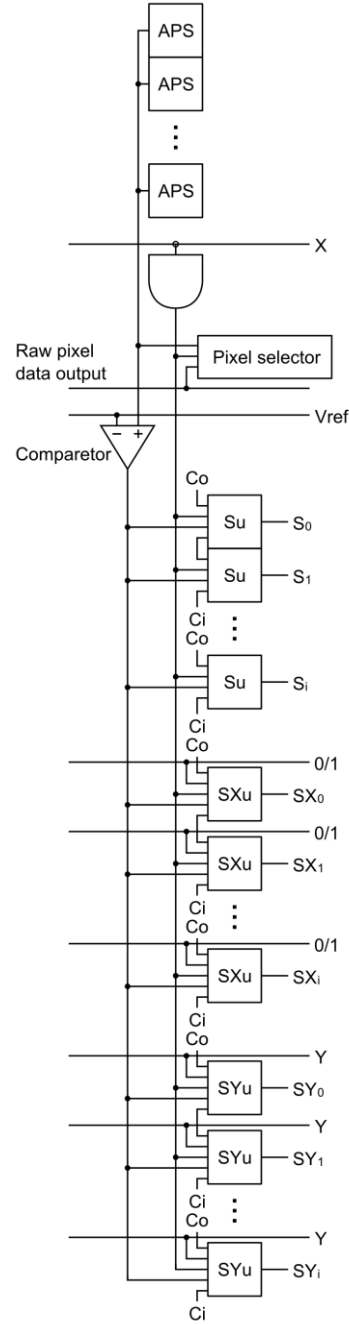


Fig.2: The circuit of column-parallel architecture for centroid calculation (for one column)

The centroid calculation is performed in parallel at each column, and the required operation frequency f is expressed as follows.

$$f = F \cdot (y + x) \quad (2)$$

Here, F , x and y are the frame rate, horizontal and vertical number of pixels, respectively, and the total number of calculation step is $(y + x)$, since the centroid calculation is performed in top to down (vertical) at first, and then left to right (horizontal) in order. The frame rate of 500[fps] and VGA resolution (640×480 pixels) gives f of 560[kHz], which can be easily achieved by the recent CMOS technologies.

The flag to indicate the pupil area, p_{xy} , is generated by a comparator with adequate threshold in each column for each sequential read-out step in entire row from standard 3-Tr pixel (APS) circuit, as shown in Fig.3.

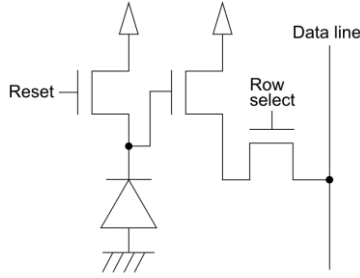


Fig.3: The circuit of CMOS active pixel image sensor

The calculation of $\sum_x \sum_y p_{xy}$ and $\sum_x \sum_y x p_{xy}$ are performed in parallel at each column. The calculation of the x - and y -direction, $\sum_x \sum_y$ is divided into y -direction (\sum_y) and x -direction (\sum_x), and (\sum_y) can be calculated at the sequential read-out step for entire rows independently in each column. After finishing read-out steps, the calculation of (\sum_x) is performed with read-out and accumulation for all the columns in order, during the next integration period.

3.1. Calculation of $\sum_x \sum_y p_{xy}$

The circuit of calculating $\sum_y p_{xy}$ at each column is composed of combined element circuit (Su), which is composed of one bit full-adder, register and buffer, as shown in Fig.4. The serial connection of Su forms a ripple-carry adder and accumulation register for the required number of bits in calculations.

The sum of the binary data from the comparator and the accumulation value is stored in the register for each row read-out step, to calculate the number of black pixels in each column.

After finishing calculation of $\sum_y p_{xy}$ in each column, the sum of the each column's result is performed by sequential read-out and accumulation for all the columns, in order to obtain the $\sum_x \sum_y p_{xy}$, the area of the pupil. The accumulation for the column direction is implemented by the adder and the register located at outside the column array, as shown in Fig.5.

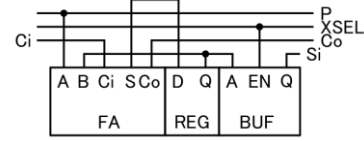


Fig.4: The circuit of $\sum_y p_{xy}$ calculation

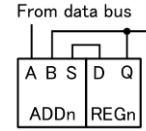


Fig.5: The circuit of column-direction accumulation

3.2. Calculation of $\sum_x \sum_y x p_{xy}$

The calculation of $\sum_x \sum_y x p_{xy}$ is similar to that of $\sum_x \sum_y p_{xy}$ described in section 3.1. Fig.6 shows the circuit of $\sum_y x p_{xy}$ calculation. The product of the x -coordinate for each column and the binary data from the comparator is accumulated with the similar architecture of $\sum_y p_{xy}$ calculation. Note that this product can be calculated by using the AND gate array, since the multiplier of x , p_{xy} , is 0 or 1. The accumulation of all the columns is also performed with the same manner as $\sum_x \sum_y p_{xy}$ by the externally located accumulator.

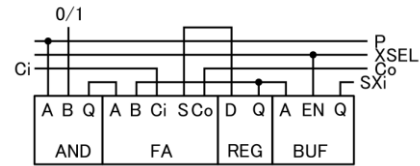


Fig.6: The circuit of $\sum_y x p_{xy}$ calculation

3.3. Calculation of $\sum_x \sum_y y p_{xy}$

The calculation of $\sum_x \sum_y y p_{xy}$ is also performed by the similar manner described in section 3.1 and 3.2.

Fig.7 shows the circuit of $\sum_y y p_{xy}$ calculation. The product of y -coordinate and the binary data from the comparator is calculated by using the AND gate array, and the sum of the products for all the column is calculated and accumulated by using adder array at each read-out step for entire row.

After the accumulation, each values of $\sum_y yp_{xy}$ are accumulated for all the columns in x -direction by the circuit located outside the column array.

The centroid of the pupil can be calculated by eq.(1) with these three values for each frame.

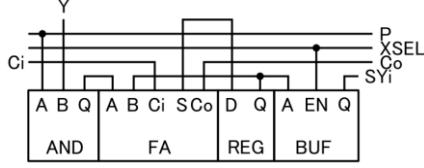


Fig.7: The circuit of $\sum_y yp_{xy}$ calculation

4. Implementation

We are now designing the LoS detection system with the processing architecture described in section 3 on two hardware platforms. One is on the emulation system composed of the high-speed camera and FPGA, as shown in Fig.8, where column-parallel processing architecture will be truly implemented for algorithm evaluation. The other is on the integrated image sensor with 128×32 pixels using standard CMOS 0.18um, five layers of metal technology.



Fig.8: The emulation system composed of the high-speed camera and FPGA

5. Prediction of Saccade Movement

The real-time tracking of the saccade movement is expected to realize the estimation (prediction) of the goal point of saccade movement by the beginning of the saccade, and the predicted goal point of saccade movement can be used a new type of information display methodology.

We carried out the measurement of saccade movement by using the high-speed camera (up to 1200[fps]) and off-line processing. Fig.9 shows an example of the measured saccade movement (x -coordinate of the pupil in eye image), and Fig.10

shows the distance of saccade movement (from the beginning to the goal) against the slope (speed) of saccade movement. From the result in Fig.10, it is expected to estimate (predict) the goal point of the saccade movement at the beginning step of saccade movement. The detailed evaluation will be reported in our future works.

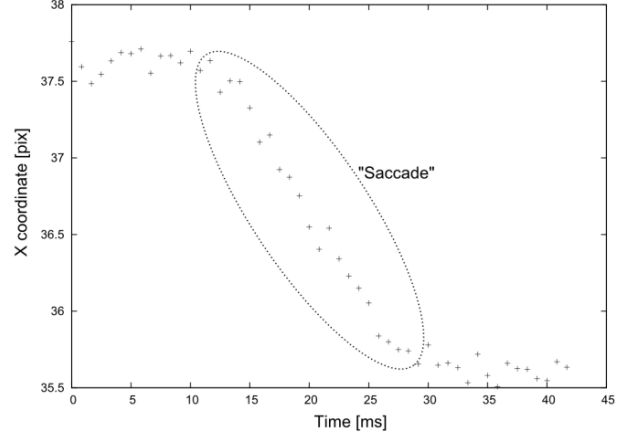


Fig.9: An example of observed saccade using high-speed camera

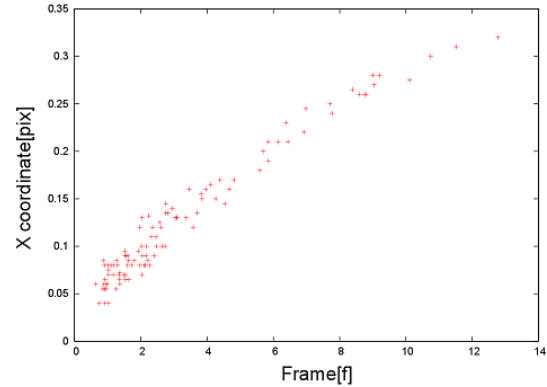


Fig.10: Distance of the saccade movement against the speed of saccade movement

6. Conclusion

We proposed the column-parallel LoS calculation architecture for integrated image sensor based on centroid calculation. We also demonstrated the evaluation of saccade prediction with the saccade movement. The implementation of an emulator, design an image sensor and estimation are now in progress work.

References

- [1] E.A.Hoffman, and J.V.Haxby, "Distinct representations of eye gaze and identity in the distributed human neural system for face perception", *Nature Neuroscience*, vol.3, pp.80–84, 2000.1.
- [2] J.Triesch et al., "What you see is what you need", *Journal of Vision*, Vol.3, No.1, pp.86-94, 2003.2
- [3] T.Takegami et al., "An Algorithm for Model-Based Stable Pupil Detection for Eye Tracking System", *Systems and Computers in Japan*, Vol.35, No.13, pp.21-31, 2004.11
- [4] J.Akita, et al., "Column-Parallel Vision Chip Architecture for High-Resolution Line-of-Sight Detection Including Saccade", *IEICE Trans. on Electronics*, Vol.E90-C, No.10, pp.1869-1875, 2007.10.