Jpn. J. Appl. Phys. Vol. 40 (2001) pp. 2585–2592
Part 1, No. 4B, April 2001

A νMOS Vision Chip Based on Cellular-Automaton Processing

Tetsuya ASAI[1,*], Tatsuhiko SUNAYAMA[1], Yoshihito AMEMIYA[1] and Masayuki IKEBE[2]
[1]*Department of Electrical Engineering, Hokkaido University, Kita 13, Nishi 8, Sapporo 060-8628, Japan*
[2]*Dai Nippon Printing Co., Ltd., Ichigaya Kagacho1-1-1, Shinjuku-ku, Tokyo 162-8001, Japan*

(Received September 11, 2000; accepted for publication November 21, 2000)

A neuron MOS (νMOS) vision chip was designed and fabricated for developing high-speed parallel image-processing systems based on cellular-automaton processing. The chip consists of cellular νMOS circuits that implement two fundamental functions in digital image processing: i) cleaning up noise in binary images and ii) detecting edges in the images, in addition to photosensing and image quantizing. Experimental results reveal that the fabricated chip successfully extracted edges from noisy inputs, which demonstrates the great potential of the νMOS vision chip in future image-processing applications.

KEYWORDS: vision chip, cellular automaton, image processing, edge detection, νMOS transistor

1. Introduction

Smart vision systems are promising components of future intelligent systems. Conventional vision systems, based on system-level integration of an imager and a digital processor, are difficult to develop into smart vision systems because of their large size and complexity. To overcome this problem, we have proposed several image-processing LSIs aiming at *chip-level integration* of the imager and processors. In this paper, we present one such LSI: namely, a neuron MOS (νMOS) vision chip based on cellular-automaton processing.

Vision chips, which include both photosensors and processing elements (pixels), have been under research for more than a decade and exhibit promising capabilities in future vision systems.[1,2] The essential ideas of the vision chips are: i) each pixel consists of a photosensor and a compact processor and ii) these processors are implemented on one chip in a fully parallel construction, which is the most important feature of the vision chip. The parallel construction ensures that the operation speed does not decline much even if the number of pixels increases. If very large-scale integration is attained, these chips will drastically improve the performance of image processing compared with conventional digital signal processors.

Vision chips can be categorized into two types: i) neuromorphic[1–3] and ii) artificial vision chips.[4,5] Neuromorphic vision chips are generally constructed by mimicking biological nervous systems and are expected to provide human-like flexible vision systems; however, there are severe difficulties in modeling nervous systems and achieving reliable processing. On the other hand, artificial vision chips, whose development and fabrication have recently been reported in the literature, certainly act as powerful visual processors due to their strict and reliable operations on the basis of existing artificial image-processing algorithms and architectures.[2,6–8]

Among various image-processing architectures, a cellular automaton (CA) is one candidate that may provide high-speed parallel image-processing systems.[9] The CA is particularly well suited for the coming generation of massively parallel machines in which a very large number of separate processors act in parallel. If a processing element in the CA is constructed from a combination of a processor and a photosensor, as in vision chips, various CA algorithms can easily be used in the development of artificial vision chips. Recently, a number of CA algorithms for extracting various image features have been proposed and successfully demonstrated,[11] showing the great potential of CA-based vision chips in image-processing applications.

Aiming at the development of CA-based vision systems, a number of CA chips have been proposed and fabricated.[12–17] Because image-processing applications need a large number of pixels, the unit processor (pixel circuit) must be constructed as compactly as possible. To meet this requirement, a functional metal-oxide semiconductor (MOS) device, known as the νMOS transistor, was recently developed and has been applied to various image-processing applications.[18,19] We developed a number of CA circuits using νMOS transistors (cellular νMOS circuits) for morphological processing on binary and gray-scale images and demonstrated that the functions required for the processing can be successfully achieved with compact pixel circuits.[20–23]

In this paper, we present a CA-based νMOS vision chip that performs noise cleaning and edge detection in binary images. In §2, we outline the CA and introduce algorithms for noise cleaning and edge detection. In §3, we show a compact CA circuit that uses νMOS transistors for variable-threshold operations. In §4, experimental results for the fabricated chip are presented.

2. Image Processing Systems Based on Cellular Automata

2.1 Pattern transformation with cellular automata

A CA is a spatially extended dynamical system with discrete time and discrete space.[9,11] It consists of a discrete lattice of *cells*, as shown in Fig. 1. Each cell carries a binary, multi valued, or continuous state.

In the CA, a square *window* region consisting of m-by-m cells is defined for each cell. For example, the window of cell $C_{i,j}$ with $m = 3$ is represented by a dashed box in Fig. 1. A window pattern is thus represented by spatial distributions of the states of a cell and its neighboring cells.

The cell states are updated synchronously in a sequence of discrete time steps according to a given *rule* and *template pattern* consisting of m-by-m pixels. The subsequent state of a cell is determined by the rule with respect to comparison results between window patterns of the cell and the template pattern. For instance, if cells are assumed to have a binary

*E-mail address: asai@sapiens-ei.eng.hokudai.ac.jp

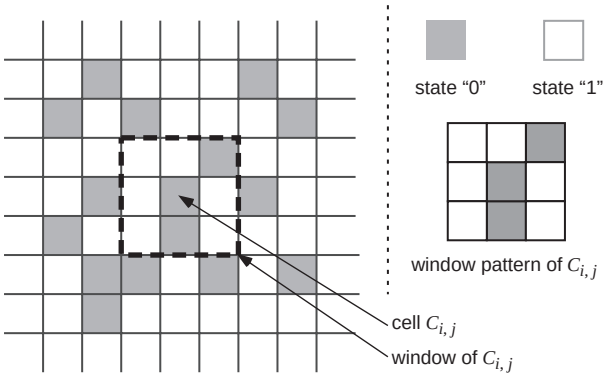


Fig. 1. CA consisting of a discrete lattice of cells. A window of a cell $C_{i,j}$ with 3-by-3 pixels is represented by a dashed box.

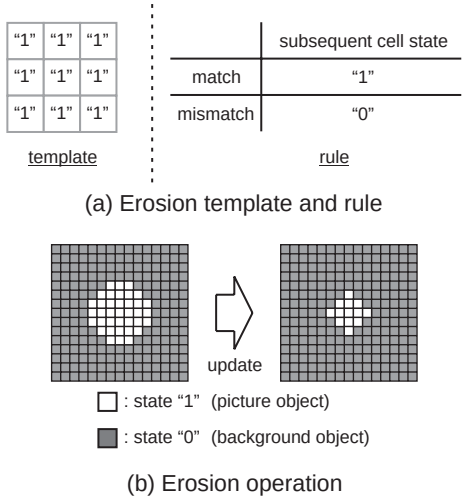


Fig. 2. (a) Erosion rule and its template with $m = 3$. (b) Example operation of erosion.

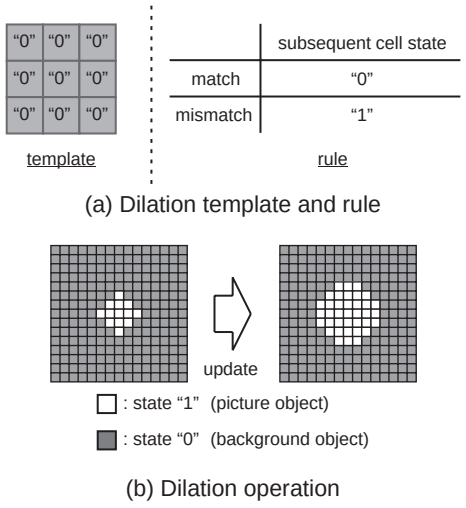


Fig. 3. (a) Dilation rule and its template with $m = 3$. (b) Example operation of dilation.

state ("0" or "1") and the window pattern of a cell matches a given template, the subsequent state of the cell is set to "1"; otherwise, it is set to "0". According to the given rules and templates, the CA produces various spatio temporal patterns in the cell state space.

2.2 Digital image processing with cellular automata

The CA can be used for various image processing applications by regarding each cell and its state as a pixel and the pixel-luminance value, respectively. In this subsection, we introduce three fundamental rules and templates for noise cleaning and edge detection.

Processing to clean up noise and detect edges is necessary for digital image-processing applications because i) the amount of noise directly affects the image quality and ii) edge information is one of the most important cues for feature extraction and image compression. In particular, in binary images, there is extensive quantization noise in addition to inherent image noise. These types of noise have to be removed before edge detection. For noise removal, we employ two rules and templates in the CA, which are called *erosion* and *dilation*.[7)]

Consider a two-dimensional image consisting of "0" (black) or "1" (white) pixels, and assume that black and white pixels represent background objects and picture objects, respectively. After an initial image has been given to a CA that consists of the same number of pixels as the image, the cell states are updated synchronously in a sequence of discrete time steps according to the given rule and templates.

An erosion rule and its template with $m = 3$ are illustrated in Fig. 2(a). With the rule and template, the CA sets the center pixel (subsequent cell state) in the window to "1" (picture object) if all the neighboring 8 cells are "1"; otherwise, the CA sets the center pixel to "0" (background object). When the rule and template are applied to all cells, an object shrinks by a single-pixel-wide ring of interior pixels after the update, as shown in Fig. 2(b). Since window patterns of cells around object edges do not match the erosion template, the states of the cells are set to "0" (background object). This means that a single-pixel-wide picture object surrounded by at least single-pixel-wide backgrounds disappears after the update.

With a dilation rule and its template with $m = 3$, the CA sets the center pixel (subsequent cell state) in the window to "0" (background object) if all the neighboring 8 cells are "0"; otherwise, the CA sets the center pixel to "1" (picture object), as shown in Fig. 3(a). In this case, a picture object grows uniformly by a single-pixel-wide ring of exterior pixels after the update (Fig. 3(b)). A single-pixel-wide background surrounded by at least a one-pixel-wide picture disappears after the update, which is the inverse operation of erosion.

The erosion and dilation operations are useful for eliminating small elements (possibly noisy) isolated from a primary element. In the case of erosion, the element is the picture object, while in dilation it is the background object. Note that the primary element, which should not be eliminated by these operations, has to be constructed by at least m-by-m pixels; otherwise, it will disappear after the update. By combining the dilation and erosion templates, we can remove noisy pixels from input images, as shown in Fig. 4. In order to keep the object size constant, the number of operations in dilation and erosion must be the same.

Finally, we introduce a rule and a template for edge detection (Fig. 5(a)). The template is the same as the erosion template, but the rule is different. With the rule and template, the CA sets the center pixel (subsequent cell state) to "0" (background object) if all the neighboring 8 cells are "1" (picture

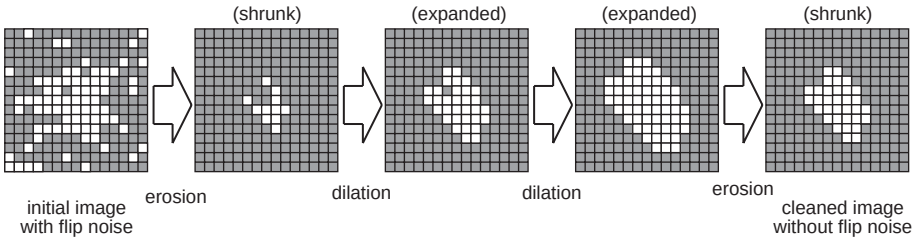


Fig. 4. Example operations of noise cleaning using dilation and erosion.

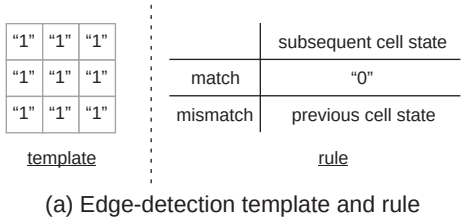


Fig. 5. (a) Edge-detection rule and its template with $m = 3$. (b) Example operation of edge detection.

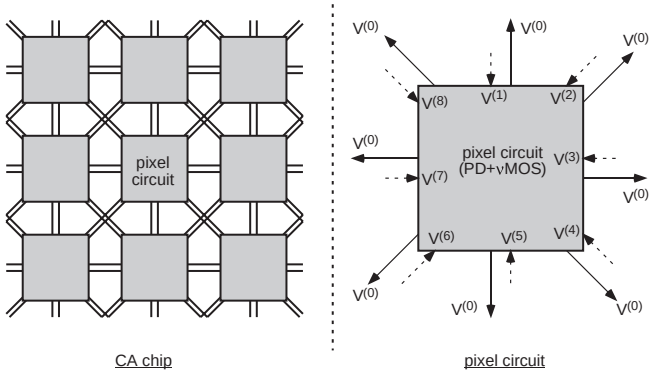


Fig. 6. Overview of the proposed vision chip, which consists of a two-dimensional array of pixel circuits.

object); otherwise, it is kept in its previous cell state. The interior pixels of a picture object are converted to "0" (background), but the peripheral pixels are not. As a result, the edge of the object is extracted, as shown in Fig. 5(b).

3. Circuit Implementation of Cellular-Automaton Image Processing System

In this section, we show cellular νMOS circuits implementing the CA for noise cleaning and edge detection. In the circuit, a silicon functional device known as the νMOSFET[18,19] is employed for a variable-threshold logic operation, so that it is suitable for implementing the CA.[20–23]

Figure 6 shows the floorplan of the νMOS vision chip consisting of a two-dimensional array of cellular νMOS circuits (pixel circuits). The chip operates in three modes: i) the photosensing mode for obtaining the initial states of the CA, ii) the CA operation mode, and iii) the readout mode. The modes are selected by external control signals, as shown below.

Figure 7 shows the cellular νMOS circuit (pixel circuit) consisting of five subcircuits: i) a photosensor, ii) cell-state memory circuits, iii) a multiplexer, iv) template-matching circuits, and v) rule selection circuits.

The photosensor consists of a photodiode (PD), a capacitor (C), an nMOS transistor (M_R), and an inverter (I0). When $V_{\text{photoinput}}$ is set to logical "1" (photosensing mode), the photosensor is connected to the cell-state memory circuit through inverters I4 and I5 and transfer gate T6 in the multiplexer. Before sensing the luminance values, the charges in capacitor C are initialized by setting V_{reset} to VDD, and V_p is set to 0 V. The sensing starts when V_{reset} is set to 0 V. The PD accepts an incident image and generates the corresponding photocurrent. The charge of the photocurrent is stored in capacitor C, and the binary-quantized data is transduced into node P. Then, the memory circuit stores the quantized data (specifying the current state of the cell) into the D-type flip flop (D-FF).

Figure 8(a) shows a primary template-matching circuit extracted from the pixel circuit in Fig. 7. The circuit consists of a νMOS inverter and an nMOS transistor (M_I) that connects a floating gate of the inverter with the output of the inverter ($\overline{V_{\text{out},\nu}}$). Additional inverters (I1 and I2) are employed as buffers, and the output of the template-matching circuit is denoted by V_{out}. The floating-gate voltage of the i-th pixel circuit ($V_{\text{fg},i}$) is given by:

$$V_{\text{fg},i} = \frac{C_0(2\sum_{j=0}^{8} V_{\text{i}}^{(j)} + 16\, V_{\text{template}} + V_{\text{initFG}}) + C_{\text{ox, p}} VDD + Q_0}{35\, C_0 + C_{\text{ox, p}} + C_{\text{ox, n}}}, \tag{1}$$

where C_0 is the unit capacitance of the νMOS inverter, Q_0 the initial charge of the floating gate, $C_{\text{ox, p(n)}}$ the gate-substrate capacitance of p(n)-type νMOS transistor, VDD the supply voltage, $V_{\text{i}}^{(0)}$ the current-state voltage in the memory circuit of the i-th pixel, $V_{\text{i}}^{(1,2,\ldots,8)}$ the current-state-voltage in the memory circuits of the neighboring pixels, V_{initFG} the control

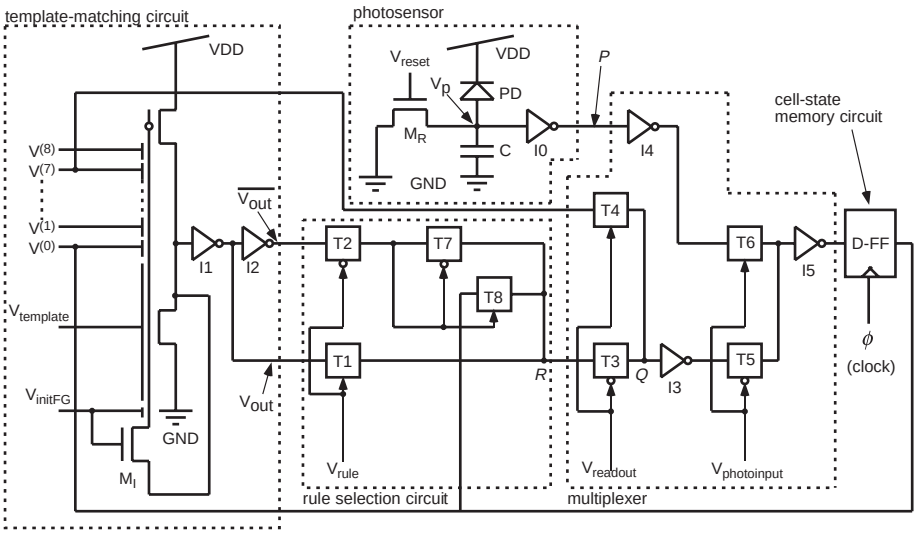
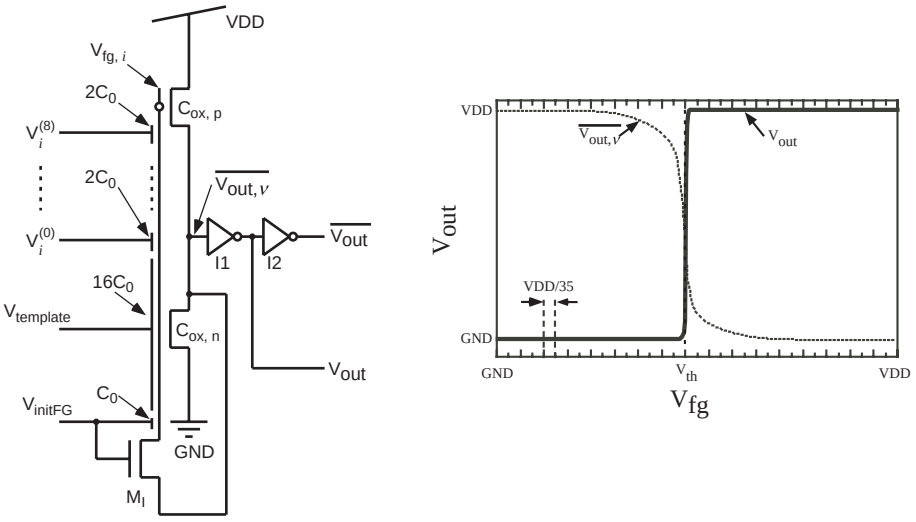


Fig. 7. Cellular νMOS circuit (pixel circuit) consisting of a photosensor, cell-state memory circuits, multiplexer, template-matching circuits, and rule selection circuits.



(a) template-matching circuit (b) input-output characteristic

Fig. 8. (a) Template-matching circuit extracted from Fig. 7. (b) Input-output characteristics of the νMOS inverter with $V_{\rm th} = {\rm VDD}/2$ after the reset operation ($V_{\rm initFG} = {\rm VDD} \rightarrow 0\,{\rm V}$).

voltage for reset operations, and $V_{\rm template}$ the control voltage for template-matching operations. When $C_0 \gg C_{\rm ox,\,p(n)}$, we obtain a simple form

$$V_{{\rm fg},i} \approx \frac{2\sum_{j=0}^{8} V_{\rm i}^{(j)} + 16\,V_{\rm template} + V_{\rm initFG}}{35} + \frac{Q_0}{35\,C_0}, \quad (2)$$

of eq. (1). Note that the νMOS inverter acts as a simple inverter that receives its input as $V_{{\rm fg},i}$.

During the reset operation of the floating gate, $V_{\rm initFG}$ is set to VDD. The floating-gate voltage ($V_{{\rm fg},i}$) is then obtained as

$$V_{{\rm fg},i} = \frac{17\,C_0\,{\rm VDD} + Q_0}{35\,C_0}, \quad (3)$$

when $V_{\rm i}^{(0,1,\ldots,8)}$ and $V_{\rm template}$ are set to 0 V and VDD, respectively. On the other hand, this voltage has to be equal to the gate threshold of the νMOS inverter because $\rm M_I$ is turned on ($V_{\rm initFG} = {\rm VDD}$), and the floating gate is connected to the output of the inverter ($\overline{V_{{\rm out},\nu}}$) through $\rm M_I$. Here the gate threshold is given by

$$V_{\rm th} = \frac{{\rm VDD} + V_{\rm tp} + V_{\rm tn}\sqrt{\beta_{\rm n}/\beta_{\rm p}}}{1 + \sqrt{\beta_{\rm n}/\beta_{\rm p}}} \quad (= V_{{\rm fg},i}) \quad (4)$$

where $V_{\rm tp(n)}$ and $\beta_{\rm p(n)}$ represent the threshold voltage and transconductance of the p(n)-type νMOS transistor. Therefore, the charge Q_0 of the floating gate after the reset operation is given by

$$Q_0 = -17\,C_0\,{\rm VDD} + 35\,C_0\,V_{\rm th}. \quad (5)$$

Figure 8(b) shows input-output characteristics of the νMOS inverter after the reset operation ($V_{\rm initFG} = {\rm VDD} \rightarrow 0\,{\rm V}$). In the figure, the gate threshold ($V_{\rm th}$) is assumed to be VDD/2. If $V_{\rm template} = {\rm VDD}$, the floating-gate voltage is ob-

tained from eqs. (3.2) and (3.5) as

$$V_{\mathrm{fg},i} = V_{\mathrm{th}} + \frac{\mathrm{VDD}}{35}(2n-1), \tag{6}$$

where n represents the number of inputs (0 to 9) receiving VDD among the inputs $V_{\mathrm{i}}^{(0,1,\ldots,8)}$. Thus, the buffered output of the νMOS inverter ($\overline{V_{\mathrm{out}}}$) becomes logical "0" when at least one input becomes logical "1" ($n > 0$). Namely, with the control voltage $V_{\mathrm{template}} = \mathrm{VDD}$, the output of the template-matching circuit (V_{out}) represents the comparison results between the current input voltages $V_{\mathrm{i}}^{(0,1,\ldots,8)}$ and the dilation template in which all pixels are logical "0", as shown in Fig. 3(a). When the current voltage distribution $V_{\mathrm{i}}^{(0,1,\ldots,8)}$, which represents the window of the i-th pixel circuit, matches the dilation template, the output of the template-matching circuit becomes logical "1"; otherwise, it is logical "0".

When $V_{\mathrm{template}} = 0\,\mathrm{V}$, the floating-gate voltage is given by

$$V_{\mathrm{fg},i} = V_{\mathrm{th}} + \frac{VDD}{35}(2n-17), \tag{7}$$

where n represents the number of inputs (0 to 9) receiving VDD among the inputs $V_{\mathrm{i}}^{(0,1,\ldots,8)}$. Thus, the buffered output of the νMOS inverter ($\overline{V_{\mathrm{out}}}$) becomes logical "0" when $n = 9$, which means all nine inputs are logical "1". With the control voltage $V_{\mathrm{template}} = 0\,\mathrm{V}$, the output of the template-matching circuit (V_{out}) thus represents the comparison results between the current input voltages $V_{\mathrm{i}}^{(0,1,\ldots,8)}$ and the erosion and edge-detection templates illustrated in Figs. 2(a) and 5(a). When the current voltage distribution matches these templates, the output of the template-matching circuit becomes logical "1"; otherwise, it is logical "0".

Now, let us return to the issues of the operation modes in the pixel circuit. In the CA operation mode, there are three submodes: i) erosion, ii) dilation, and iii) edge detection. These modes are switched by setting a control voltage V_{rule} for rule selection between the erosion and edge detection rules, and by setting the control voltage V_{template} that determines template patterns. These rules are selected by the rule selection circuits shown in Fig. 7. When V_{template} and V_{rule} are set to logical "1", the voltage on node R represents the subsequent cell states for the dilation operation because the transfer gate T1 is turned on and T2 is turned off. When V_{template} and V_{rule} are set to logical "0" and "1", respectively, the voltage on node R represents the subsequent cell states for the erosion operation. Then, if V_{template} and V_{rule} are set to logical "0", the voltage on node R represents the subsequent cell states for the edge detection because T1 is turned off and T2 is turned on. When the current state matches the template, $\overline{V_{\mathrm{out}}}$ is set to logical "0" and is transferred to the node R because T7 is turned on and T8 is turned off. If the current state does not match the template, $\overline{V_{\mathrm{out}}}$ is set to logical "1" and the current state of the cell $V^{(0)}$ in the memory circuit is transferred to node R because T7 is turned off and T8 is turned on. This is the same rule as for edge detection illustrated in Fig. 5(a).

In the CA operation mode, V_{readout} and $V_{\mathrm{photoinput}}$ are set to logical "0". The subsequent cell states for the selected rules and templates are thus transferred to the memory circuit through transfer gates T3 and T5 and inverters I3 and I5. The cell state in the memory is updated with a clock signal (ϕ).

In the readout mode for retrieving the processed data, control signals V_{readout} and $V_{\mathrm{photoinput}}$ are set to logical "1" and "0", respectively. Then, a set of pixel circuits in the same row operates as a shift register consisting of D-type flip flops in the memory circuits. The pixel circuit receives the pixel data from the left-hand pixel circuit through input terminal $V^{(7)}$, as shown in Fig. 6. The received data is then given to the memory circuit through transfer gates T4 and T5 and inverters I3 and I5. Then, the data is passed synchronously to the right-hand pixel circuit with the clock signals (ϕ). The processed image data on each row of pixel circuits is obtained in the form of a 1-0 bit stream, that is, the output from the right-end cell.

4. Experimental Results

The CA chip was fabricated by a 0.6-μm double-poly triple-metal CMOS process. Figure 9 shows the chip photograph implementing 15-by-15 pixel circuits. The unit cell circuit was 210 μm × 200 μm in size. In the following experiments, the supply voltage (VDD) was set to 3.3 V.

First, we confirmed the storage operations of the photosensor and cell-state memory circuit. Ambient room light was incident upon the upper part of the chip, while the lower part was covered with metal masks, as shown in Fig. 10(a). With all the pixel circuits set to the photosensing mode $V_{\mathrm{photoinput}} = \mathrm{VDD}$, the quantized binary data was given to each memory circuit in the pixel. A clock of 1 kHz was applied to the reset terminal (V_{reset}), so the storage time of the photosensor was set to 0.5 ms. At the end of the storage operation, a single pulse was applied to the CLK terminal to store the binary data in the memory circuit. After storage, the pixel circuits were set to the readout mode, that is $V_{\mathrm{readout}} = \mathrm{VDD}$ and $V_{\mathrm{photoinput}} = 0\,\mathrm{V}$. Then, a 1 MHz clock was applied to all the memory circuits. Figure 10(b) shows the readout results. Since the pixel data of each row is obtained in the form of a 1-0 bit stream by the shift resisters, the x and y axes represent the position of the pixel circuits, and values (logical "0" or "1") at position (x, y) represent the memory states. The incident image data on each row of pixel circuits was successfully retrieved by the storage and readout operations.

Figure 11 shows results for the basic operations of dilation, erosion, and edge detection. The initial image is shown in Fig. 11(a). First, we set V_{initFG} to VDD to reset all the float-

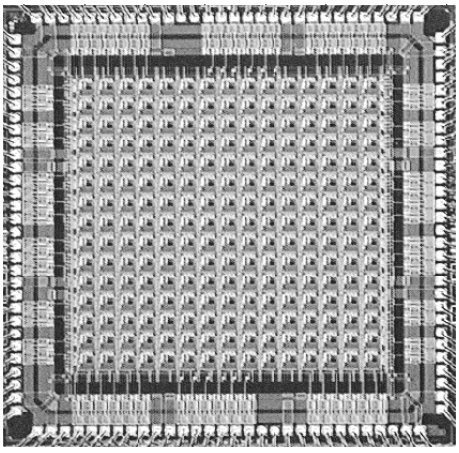


Fig. 9. Chip micrograph. The chip was fabricated in a 0.6-μm double-poly triple-metal CMOS process.

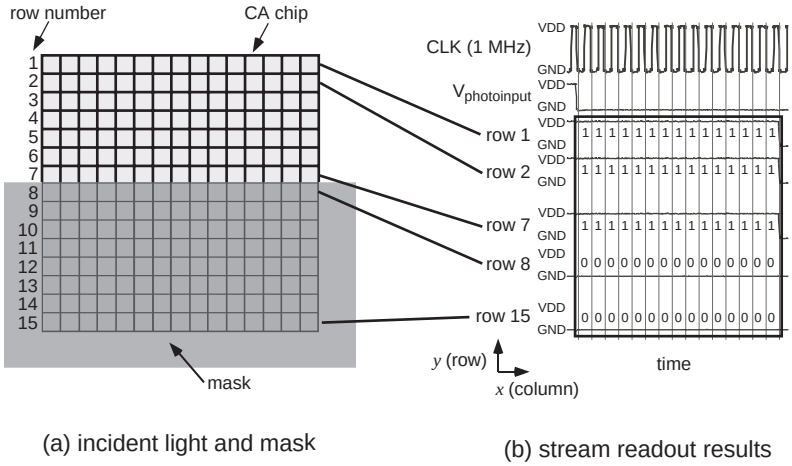


Fig. 10. Results of photosensing experiments. (a) Photomask setting and (b) readout results from the chip without the CA operations.

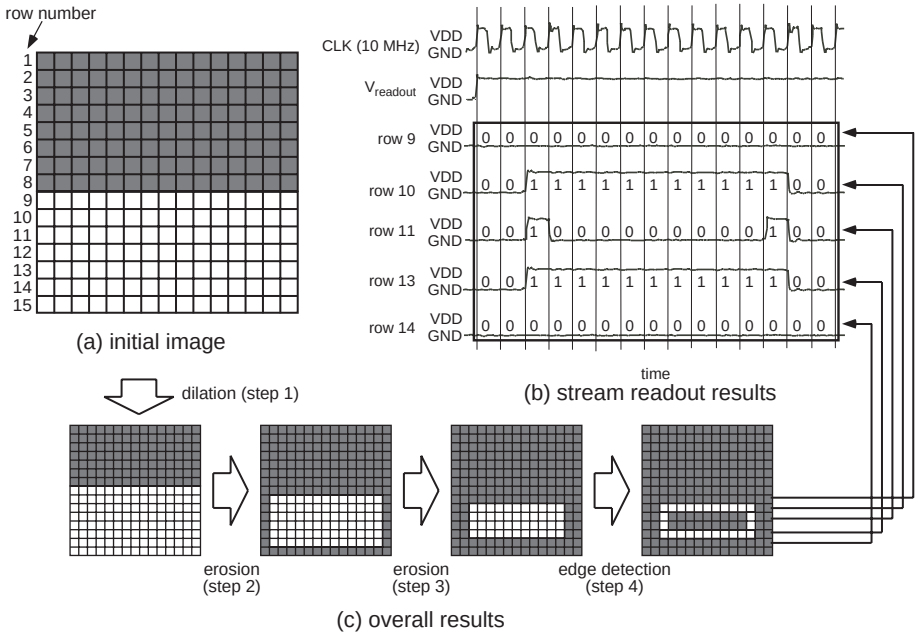


Fig. 11. Results of basic CA experiments. (a) Initial image, (b) readout results from the chip after the CA operations, and (c) overall results.

ing gates. Then, in the first step (step 1), V_{initFG}, V_{template}, V_{rule}, V_{readout}, and $V_{\mathrm{photoinput}}$ were set to logical "0", "1", "1", "0", and "0", respectively, that is, the CA operation mode with the dilation submode. In the next two steps, V_{template} and V_{rule} were set to logical "0" and "1", respectively, that is, the erosion submode. Finally, V_{template} and V_{rule} were set to logical "0", that is, the edge detection submode. After these operations, the pixel circuits were set to the readout mode. Figure 11(b) shows the readout results. The processed image (edge) data on each row of pixel circuits was successfully retrieved by the erosion, dilation, edge detection, and readout operations. Overall readout results are shown in Fig. 11(c). In the first step, the initial image was dilated by one upper row. In the next two steps, the dilated image was eroded by two inner rings. In the fourth (final) step, the edge was extracted from the eroded image. In this experiment, the chip was able to operate at a clock frequency of up to 10 MHz.

We confirmed the operations of the CA chip for practical images including extensive noise. The letter "D" was chosen as the primary image [Fig. 12(a)]. We generated a noisy "D" from the primary image, as shown in Fig. 12(b), and then applied the noisy "D" to the CA chip as an initial state. After resetting the floating gate, we operated the chip in the sequence of erosion (step 1), dilation (step 2), dilation (step 3), erosion (step 4) and edge detection (step 5). Figures 12(c) and 12(d) show the overall results and the detailed readout results, respectively. This noisy "D" was cleaned up within four clock intervals through dilation and erosion processing (steps 1 to 4), then the edge of the cleaned image was extracted (step 5). The final data on each row of pixel circuits was successfully retrieved. In this experiment, the chip was able to operate at a clock frequency of up to 1 MHz.

The total power consumption of the CA chip becomes maximum when floating-gate voltages of νMOS inverters are equal to their gate-threshold voltages ($V_{\mathrm{fg}} = V_{\mathrm{th}}$). This state occurs when black and white pixels are alternately arranged in the cell space, i.e., a *checkerboard pattern* is given to the chip. In this state, the number of logical "0" inputs of a νMOS

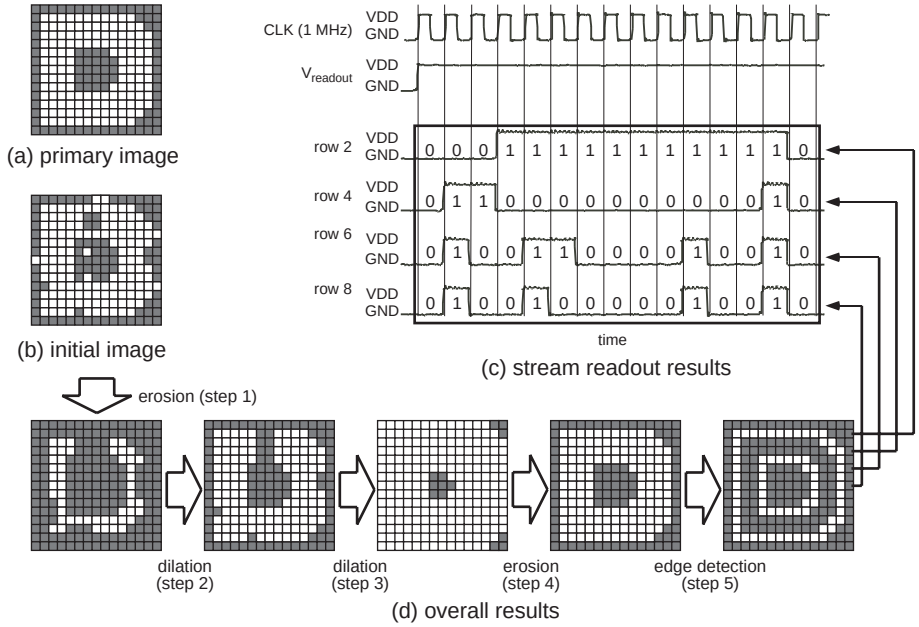


Fig. 12. Results of practical CA experiments. (a) Primary image (letter "D"), (b) initial image, (c) readout results from the chip after the CA operations, and (d) overall results.

inverter is equal to the number of logical "1" inputs of the inverter. Notice that this state results in the worst operation of the chip in terms of power consumption. We assumed that the checkerboard pattern was given to the chip, and measured the total power of the chip by setting V_{initFG} of all the cell circuits (15×15 circuits) to logical "1" ($V_{\mathrm{fg}} = V_{\mathrm{th}}$). This operation is equivalent to applying checkerboard patterns to the chip. The resultant power consumption was approximately 307 mW (1.4 mW per cell circuit).

In the above experiments, the maximum clock frequencies were quite low (10 MHz for basic operations and 1 MHz for operations on the letter "D") compared with conventional digital chips, but they are sufficient for real-time image processing because the CA operation is inherently parallel and the required clock frequency does not depend on the number of pixels. Indeed, in the proposed νMOS vision chip, noise cleaning and edge detection require only five clock intervals (steps in the CA) without the photosensing, reset, and readout operations, independently of the image's pixel size. Namely, when the chip operates at a 1 MHz clock frequency, the resultant operational frame rate is 200 kfps, which is extremely high compared with those of commercially available products.

5. Conclusions

A νMOS vision chip was designed and fabricated for developing high-speed parallel image-processing systems based on cellular-automaton processing. Experimental results showed that the fabricated chip successfully captured incident images, cleaned up noise in the images, and extracted edges from the cleaned images. For practical image processing, the maximum clock frequency was 1 MHz, which indicates that the chip can perform noise cleaning and edge detection at 200 kfps. The results show the great potential of the νMOS vision chip in future image-processing applications.

Acknowledgement

This work was supported by a Grant-in-Aid for Scientific Research on Priority Areas [(A)(2)12044201] from the Ministry of Education, Science, Sports and Culture and by the Semiconductor Technology Academic Research Center (STARC).

1) C. Mead: *Analog VLSI and Neural Systems* (Addison Wesley, New York, 1989) 1st ed.
2) C. Koch: *Vision Chips: Implementing Vision Algorithms with Analog VLSI Circuits* (IEEE Computer Society Press, Los Alamitos, 1995).
3) T. S. Lande: *Neuromorphic Systems Engineering: Neural Networks in Silicon* (Kluwer Academic Publishers, 1998) 1st ed.
4) K. Kyuma: Nature **372** (1994) 197.
5) T. Komuro, I. Ishii and M. Ishikawa: Advanced Robotics **12** (1999) 619.
6) T. Poggio, W. Yang and V. Torre: *The Computing Neuron* (Addison Wesley, New York, 1989) p. 355.
7) W. K. Pratt: *Digital Image Processing* (John Wiley & Sons, New York, 1991) 2nd ed.
8) A. Rosenfeld and A. C. Kak: *Digital Picture Processing* (Academic Press, San Diego, 1982) 2nd ed.
9) T. Toffoli and N. Margolus: *Cellular Automata Machines* (MIT Press, Cambridge, 1987)
10) A. Adamatzky: *Identification of Cellular Automata* (Taylor and Francis, London, Bristol, 1994)
11) K. Preston, Jr. and M. J. B. Duff: *Modern Cellular Automata: Theory and Applications* (Plenum Press, New York, 1984) p. 17.
12) E. Macii and M. Poncino: IEEE Trans. Reliab. **46** (1997) 173.
13) S. Nandi, S. Chattopadhyay and P. P. Chandhuri: Proc. 9th Int. Conf. on VLSI Design (1995) p. 61–4.
14) E. D. Adamides, P. Tsalides and A. Thanailakis: Proc. 6th Mediterranean Electrotechnical Conf. (1991) p. 335–8.
15) J. E. Varrientos, J. R. Angulo and E. S. Sinencio: Proc. 33rd Midwest Symp. on Circuits and Systems (1991) p. 277–80.
16) P. Tsalides, A. P. Marriott, P. J. Hicks and A. Thanailakis: Proc. VLSI and Computer Peripherals. VLSI and Microelectronic Applications in Intelligent Peripherals and their Interconnection Networks (1989) p. 3–99.
17) D. Sawh, J. Loewen, W. Lehn, H. C. Card, M. Pawlak, D. M. Burek and R. D. McLeod: Tech. Dig. Proc. Canadian Conf. on Very Large Scale Integration (1986) p. 133–8.
18) T. Shibata and T. Ohmi: IEEE Trans Electron Devices **382** (1992) 1444.
19) S. Jung, R. Thewes, T. Scheiter, K. F. Goser and W. Weber: IEEE J.

Solid-State Circuits **34** (1999) 978.
20) M. Ikebe, K. Kameishi and Y. Amemiya: Electri. Eng. Jpn. **126** (1999) 41.
21) M. Ikebe, M. Akazawa and Y. Amemiya: Computers & Electri. Eng. **23** (1997) 439.
22) M. Ikebe and Y. Amemiya: Proc. Int. Symp. on Future of Intellectual Integrated Electronics (1999) p. 377.
23) T. Sunayama, M. Ikebe, T. Asai and Y. Amemiya: Jpn. J. Appl. Phys. **39** (2000) 2278.