

A quadrilateral-object composer for binary images with reaction–diffusion cellular automata

TETSUYA ASAI*, MASAYUKI IKEBE, TETSUYA HIROSE and
YOSHIHITO AMEMIYA

Graduate School of Information Science and Technology, Hokkaido University, Kita 14, Nishi 9,
Sapporo 060-0814, Japan

We will describe a cellular-automaton (CA) LSI architecture that extracts quadrilateral objects, such as box areas filled with the same pixel values, from binary images. We propose a serial combination of parallel CA algorithms, based on the reaction–diffusion (RD) chemical systems model. Each cell in the CA is implemented by a simple digital circuit called an elemental processor. The CA LSI can be constructed by a large number of elemental processors and their controllers operating in serial and parallel. When we assumed a 0.25 μm complementally-metal-oxide-semiconductor (CMOS) process with the proposed circuits, the estimated total area of the elemental processor was approximately 15 by 15 μm^2 .

Keywords: Reaction–diffusion system; Reaction–diffusion chip; Non-standard computing; Cellular automata; Image processing

1. Introduction

Natural systems give us examples of amorphous, unstructured devices, capable of fault-tolerant information processing, particularly with regard to the massive parallel spatial problems that digital processors have difficulty with. For example, reaction–diffusion (RD) chemical systems have the unique ability to efficiently solve combinatorial problems with natural parallelism [1]. In liquid-phase parallel RD processors (RD chemical computers), both the data and the results of the computation are encoded as concentration profiles of the reagents. The computation is performed via the spreading and interaction of the wave fronts. In experimental chemical processors, data are represented by local disturbances in the concentrations, and computation is accomplished via the interaction of waves caused by the local disturbances.

The RD chemical computers operate in parallel because the chemical medium's micro-volumes update their states simultaneously, and the molecules diffuse and react in parallel. We see a similar parallelism in cellular automata (CA). Various RD systems can be modeled in terms of CA, including the Belousov–Zhabotinsky (BZ) reaction [1,2], the Turing system [3], a precipitating BZ system for computation of Voronoi diagram [4,5], and

*Corresponding author. Email: asai@sapiens-ei.eng.hokudai.ac.jp, URL: <http://sapiens-ei.eng.hokudai.ac.jp/>

so on. A 2-D 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 an elemental processor in the CA is constructed from a smart processor and photosensor, various CA algorithms can easily be used to develop intelligent image sensors.

Semiconductor RD computing LSIs (chips) implementing RD dynamics have been proposed in the literature. These chips were mostly designed by digital, analog, or mixed-signal complementally-metal-oxide-semiconductor (CMOS) circuits of cellular neural networks (CNNs) or CA. Electrical cell circuits were designed to implement several CA and CNN models of RD systems [6–12], as well as fundamental RD equations [13–17]. Each cell is arranged on a 2-D square or hexagonal grid and is connected with adjacent cells through coupling devices that transmit a cell's state to its neighboring cells, as in conventional CAs. For instance, an analog–digital hybrid RD chip [7] was designed for emulating a conventional CA model for BZ reactions [2]. A precipitating BZ system for computation of Voronoi diagram [4,5] was also implemented on an analog–digital hybrid RD chip [12]. A full-digital RD processor [9] was also designed on the basis of a multiple-valued CA model, called *excitable lattices* [1]. An analog cell circuit was also designed to be equivalent to spatial-discrete Turing RD systems [15]. A full-analog RD chip that emulates BZ reactions has also been designed and fabricated [14]. Furthermore, blue-prints of non-CMOS RD chips have been designed; i.e. a RD device based on minority-carrier transport in semiconductor devices [13] and a single-electron RD device [18].

In this paper, first, we consider image processing on chemical RD processors with complementary object reduction and enlargement operations. Then, we expand the idea into the CA that detects quadrilateral objects, such as box areas filled with the same pixel values, from binary images. Finally, we propose serial and parallel architectures to implement the proposed system on LSIs, and show its performance by numerical simulations.

2. Object scaling on chemical processors

Chemical reactions are formulated in terms of temporal differences in the concentrations of the chemical reagents, at specific *points* in the reaction space. If the spatial distribution of the chemicals is not uniform, the reagents will diffuse according to the reagent concentration gradient. This diffusive–reaction system, with multiple chemical reagents, is the primary component of chemical processors today. According to a basic RD system model, it is clear that chemical processors can be regarded as 2-D arrays of excitable units. In the BZ chemical processor [19], generally, three circulative states are introduced according to each unit's dynamical phase; i.e. resting state (depletion in the Br^- ion), excitable state [autocatalytic increase in the HBrO_2 ion (oxidation of the catalyzer)], and refractory state [depletion in the Br^- ion (reduction of the catalyzer)]. When a unit is resting, it is easily excited by the neighboring units and external stimuli. Then, it turns to the refractory state. During the refractory state, the unit cannot be excited even with external stimuli.

Figure 1 illustrates basic ideas of object scaling in an excitable RD processor. Here the refractory period for each unit is assumed to be infinity. When an object (background) is defined by the black (white) area, on which each unit is in the resting (refractory) state, the black area will be reduced as the time increases, because units in resting states can easily be excited by the surrounding excited units that exist along the boundary between the object and

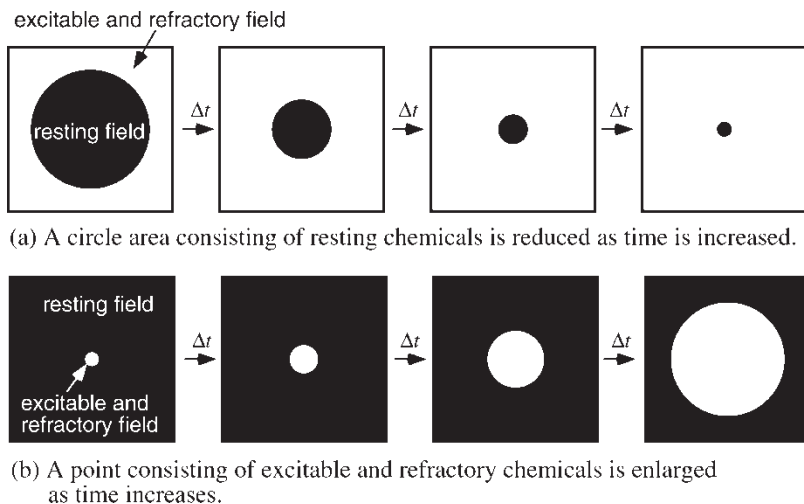


Figure 1. Image processing (scaling) in excitable chemical processors; (a) reduction of a circle (erosion of black area) and (b) enlargement of a point (dilation of white area).

background, as shown in figure 1(a). This process is called *erosion* in the chemical processor. Conversely, a *dilation* process occurs when an object (background) is defined by a white (black) area, on which each unit is in the refractory (resting) state, as shown in figure 1(b).

Let us consider a task picking up an object among multiple objects in an incident image, in order of the object size. Figure 2 shows a possible sequential algorithm for a RD processor. First, the initial input (i) in figure 2 is incident upon the processor. The input includes two circular objects, where each object is defined by a black (resting) area. As the time increases, the size of each object is reduced on the processor due to erosion. Now assume that the small circular object is about to disappear at $t = \Delta t_1$ [(ii) in figure 2], while the largest one is about to disappear at $t = \Delta t_1 + \Delta t_2$ [(iii) in figure 2]. At (iii), the shape of the circle is left unchanged, as long as the resolution of the unit array is higher than the display resolution. This is thus the smallest and most efficient representation of the circular object in

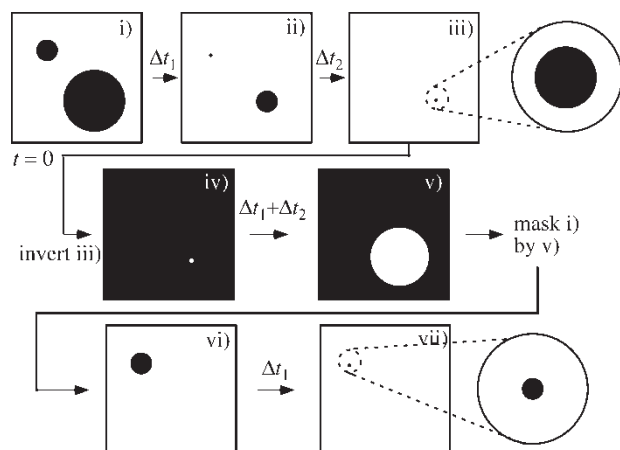


Figure 2. Object scaling in a chemical processor. An excitable chemical processor performs erosion and dilation with infinite (but molecular level) resolution.

the RD processor. So we can consider the RD processor to be a video memory that has infinite (but molecular level) resolution. Consequently, at (iii), only one circular object (the larger one) exists on the RD processor, which indicates that we can easily pick up the object, by just memorizing the coordination of the small circle (point) on the processor as well as $\Delta t_1 + \Delta t_2$. To restore the original object, we simply invert image (iii) in figure 2, which results in (iv). At this point, the RD processor automatically starts performing dilation. When $\Delta t_1 + \Delta t_2$ have passed since the start of dilation, the circular object is restored in the processor to its original size [(v) as shown in figure 2]. To pick up the small circle object in (i), we perform further erosions with the initial inputs (vi) obtained from the mask of (i) by (v). When Δt_1 have passed since the start of erosion, we see the same results; i.e. the smallest and most efficient representation of the circular object, as seen in the star object (iii). So by memorizing the coordination of the point on the processor and Δt_1 , we can pick up the small object. For large number of objects in an incident image, by just repeating the process above, we can obtain a list of a pair of the point coordination and the time, which represents a compressed representation of all of the circular objects.

3. CA for quadrilateral-object extraction

Based on the memory model of chemical computers described in the preceding section, we propose a sequential CA system that is suitable for LSI implementation. For CA implementation, we have to consider that CA's spatial resolution is not infinite, unlike chemical processors. For instance, during erosion, original shapes of objects will significantly be changed due to the finite resolution of CA operations. In this paper, we limited the shape of the objects to "quadrilateral objects" by using seven asymmetric templates and one symmetric template. Figure 3 shows all the templates where the center cell is represented by a gray box.

By using CA templates shown in figure 3, we show quadrilateral-object extraction on a sequential CA. Figure 4 shows the example operation. First, a binary input image is given to the CA [step 0 in figure 4(a)]. Then the CA performs erosion repeatedly with "erosion I" template shown in figure 3(a). At step 3 in figure 4(a), two objects (both represented by clusters of "1") are about to disappear. Subsequently, we can estimate the smallest object type in the reduced image (step 3) by using four filters; i.e. "x", "y", "xy1" and a "box", as shown

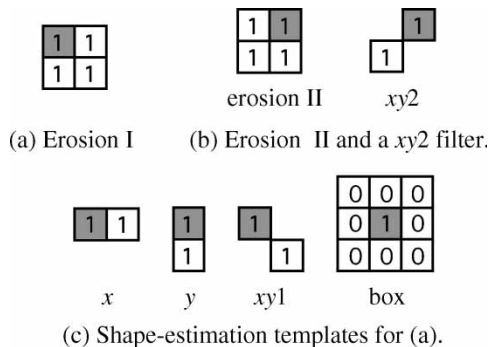
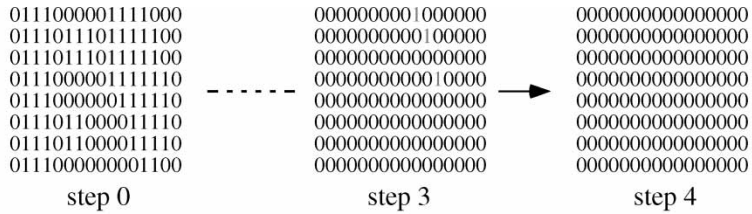
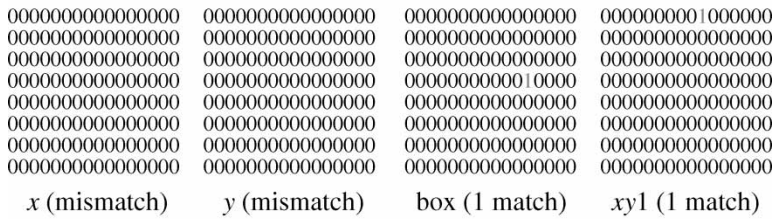


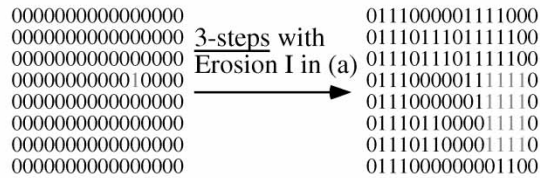
Figure 3. Basic templates for proposed CA.



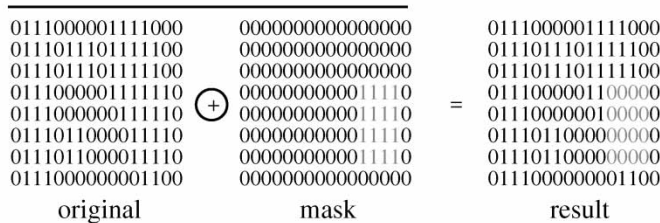
(a) Convolute initial values with template "erosion I" until all cell states = 0.



(b) Convolute a step 3 image (a) with template "x", "y", "box", "xy1".



(c) Estimate original rectangular unit form the matched box in (b).



(d) Mask the original image with the estimated box in (c).

Figure 4. Example operations of proposed CA: (a) initial erosion; (b) form detection; (c) maximum quadrilateral-object composition and (d) masking initial image by a maximum object.

in figure 3(c). In this example, the "xy1" and "box" templates can be matched [Figure 4(b)]. If no templates can be matched, the smallest object must match template "xy2", as seen in figure 3(b). In this case, we have to restart the erosion process using the "erosion II" template shown in figure 3(b).

Once the smallest object type is determined, we can estimate the maximum "box" area that can be stored within the original object, from the number of steps in the applied template, as shown in figure 4(c). At this point, we must memorize the object's box information. It should be noticed that this box is the largest box in the initial image [step 0 in figure 4(a)]. To find the second largest box, the initial image is masked with this box, either as a result of dilation, or by simply recalling the box information, as shown in figure 4(d).

4. System implementation

Figure 5 shows a block diagram of the whole CA system, which consists of four blocks: (i) a 2-D cell-circuit array, (ii) flag collection circuits, (iii) template-matching circuits, and (iv) a controller.

The cell-circuit array acts as an erosion processor for extracting quadrilateral objects, and a dilation processor for reconstructing objects that have been removed from the initial image. In the above operations, all cells work synchronously as CA. The controller manages these blocks and creates quadrilateral-object information from the erosion time steps and the results of the template matching.

Flag collection circuits that consist of OR circuits determine the condition when all cell states are at a logical “0”. The template-matching circuits, with three line buffers and template circuits, perform pattern matching on each row of the cell-circuit array at each time step. Because a matching process with “x”, “y”, “xy1”, “xy2”, and “box” templates is performed only once during the output period, we separated the erosion (dilation) CA processor from the template matching CA circuits with the line buffer.

Figure 6 shows a cell circuit with four sub-circuits: (i) crossbar switches for changing the input signals, (ii) AND and OR circuits for erosion and dilation templates, (iii) multiplexers for changing the direction of the data transfer, and (iv) cell-state-memory (flip-flop) circuits. By using signals from the controller ($\phi_1, \phi_2, \text{sel-sig0} \rightarrow \text{sel-sig2}$), these sub-circuits can operate in three modes: (i) retrieval of the initial image mode, (ii) an operating-erosion and -dilation CA mode, and (iii) a data output mode.

When the sel-sig1 is set to logical “0”, ϕ_0 supplies a CLK signal that starts from the logical “0”, and ϕ_1 is set to a logical “0”. The cell circuits for each row also act as shift registers. If the initial data is put into the left side of the cell arrays in figure 5, the memory circuits of all the cells can obtain the initial data sequentially. After this operation, when ϕ_0 and ϕ_1 are set to logical “1”, the initial data can be stored. When sel-sig0 is set to “0” (or “1”), and ϕ_0 receives a CLK signal, the cell circuit performs erosion (or dilation) on the CA cell. To obtain an instant when the cell

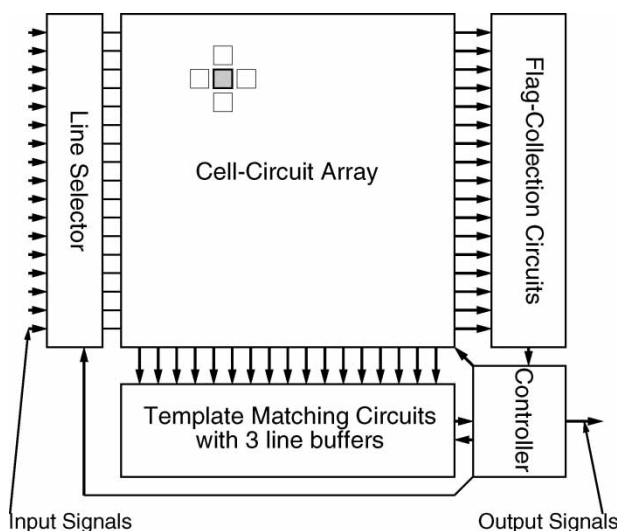


Figure 5. Block diagram of a quadrilateral object composer for binary images with reaction-diffusion cellular automata.

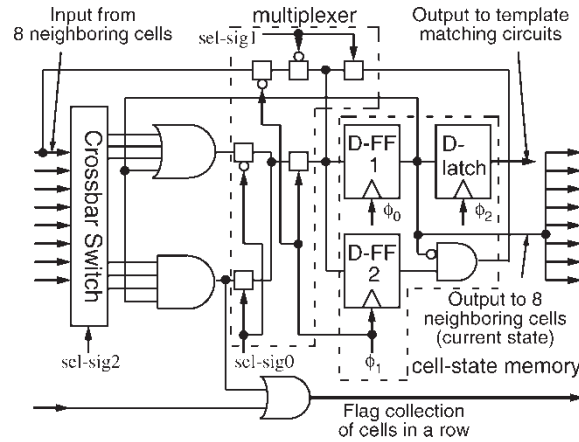


Figure 6. Cell circuit with four sub-circuits.

state are all “0”, the controller recognizes each subsequent cell state as a flag at each time step. By setting ϕ_0 to “0”, the current cell state is stored in the D-latch for output to the template circuits. We use sel-sig1 and ϕ_1 for logical operations between D-FF1 and D-FF2. This operation removes objects from the initial image, while sel-sig2 controls the crossbar switches that can change the template window.

The system has five operation periods. In the first period, the initial image is loaded in the cell array by shift registers. In the second period, the system erodes the cell array and decides whether the subsequent state of all cells is “0”. When all cell states are “0”, the third period starts. The system reads the current states of all cells at each line and performs template matching. From the results of this operation and the erosion times, information about the objects can be obtained. During the fourth period, the cell array dilates the resultant states to restore the object size. Then the objects are removed from the initial image by logical operations between the memory circuits of the cell. If the system cannot extract any objects by repeating the first to fourth period, the fifth period is processed once and the sequence of the above four periods is repeated. If the system still cannot extract any objects, the system operations are completed.

We designed a LSI layout of single cell circuit by assuming $0.25\ \mu\text{m}$ 1-poly 5-metal n -well scalable CMOS process (MOSIS, Vendor: TSMC). Figure 7 shows the layout, including the logic circuits and pass gates in figure 6, as well as a photo detector (simple pn junction between p -substrate and n -diffusion) and additional switching circuits for reset and readout operations. All circuit areas except for photodetectors were masked by top metal. The resulting cell size was approximately 15 by $15\ \mu\text{m}^2$ ($125\ \lambda \times 125\ \lambda$, $\lambda = 0.12\ \mu\text{m}$).

5. Simulation examples

We conducted Verilog HDL simulations [20] for binary images. In the simulations, all pass gates in figure 6 were replaced with conventional logic circuits. The operation flow is as follows:

- (1) Transfer the input binary image to the cell-circuit array and repeatedly perform “erosion I”. The number of erosion process is counted, and the number at which all the cell values

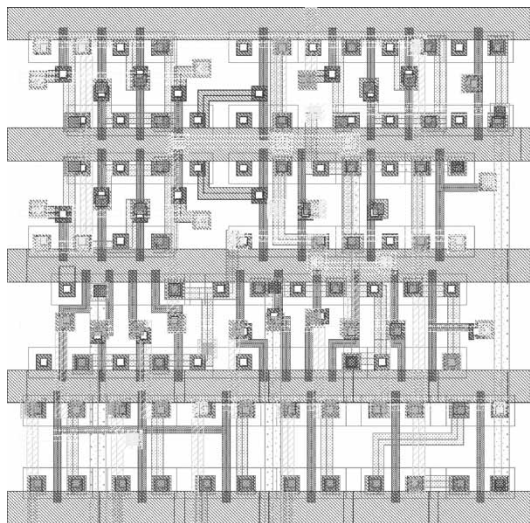


Figure 7. Single cell layout on LSI.

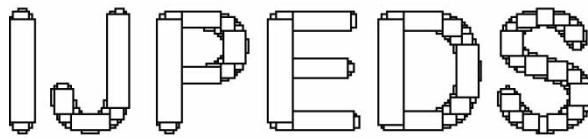
become zero ($\equiv s_c$) is memorized by the peripheral controller. Note that the D-FF1 stores the cell values just before the moment when all the cell values become zero.

- (2) Transfer the cell values in rows of three to the template matching circuits and sequentially perform x -, y -, $xy1$ - and box-template matching. The number of matching processes is counted, and the number at which all the cell values become zero ($\equiv x_c, y_c, xy1_c$ and box_c) is stored in the matching circuits.
- (3) When the row values do not match the above templates, the cell-circuit array starts the erosion process again by repeating template “erosion II”. The number of processes is counted, and the number at which all the cell values become zero ($\equiv is_c$) is memorized by the peripheral controller. Then, perform $xy2$ -template matching, as described in the preceding process (3).
- (4) Generate a quadrilateral mask by (i) the position of the resting active cell (“1”), and (ii) the values of $x_c, y_c, xy1_c$ and box_c , and create subsequent binary input images, masked by this quadrilateral box.
- (5) Repeat tasks (1)–(4) until s_c or is_c becomes zero.
- (6) Finally, produce the quadrilateral objects (vertical and horizontal rectangles and squares) from the resting active-cell position and the values of x_c, y_c , and the box_c .

Figure 8(b) illustrates the result for input bitmap image shown in figure 8(a). The input image that consisted of 210×45 pixels was decomposed of 197 quadrilateral objects. The bitmap image occupied 9450 bits (1182 bytes) on memory, while the objects used 788 bytes (8-bit address of 4 corners \times 197 boxes). The output contains only rectangles and squares, since slanted objects were removed during the last process (6) described above. Also we examined object extraction for a natural image shown in figure 9(a). The image was quantized, and given to the CA LSI. Figure 9(b) shows the results. The maximum boxes were correctly detected in order, as predicted. The input bitmap image that consisted of 181×238 pixels was decomposed of 1020 quadrilateral objects. The bitmap image occupied 43078 bits (5385 bytes) on memory, while the objects used 4080 bytes (8-bit address of

IJPEDS

(a) original binary image



(b) resultant image

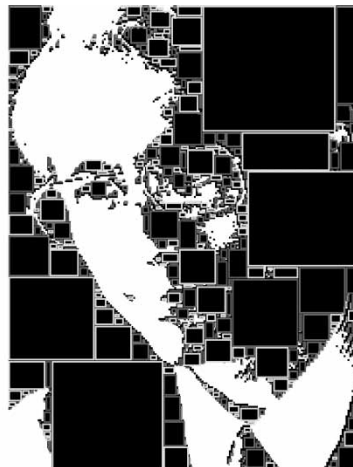
Figure 8. Simulation results for 210×45 image; (a) quantized input image given to CA LSI; (b) detected quadrilateral objects. The maximum boxes are detected in order.

4 corners \times 1020 boxes). The important thing is not to discuss the compression rate between bitmap images and extracted objects, but that bitmap images were represented by small number of vector objects, which facilitates picture drawing in terms of the drawing speed if we have variable box window.

One of the most important application targets for the proposed chip is a CAD system for VLSIs. Conventional VLSI CAD tools use polygons to represent device structures. However, recent VLSIs include not only polygon patterns but also graphical patterns, consisting of large number of dots, usually imported from image files such as JPEGs, to implement complex analog structures. In the mask manufacturing process, exposing a large number of dot patterns is quite a time-consuming task. Recently, electron beam (EB) lithography systems that can expose wide areas through a quadrilateral window



(a) original gray-scale image



(b) resultant image

Figure 9. Simulation results for 181×238 image: (a) original image; (b) mixed image of quantized image given to CA LSI and detected quadrilateral objects. The maximum boxes are detected in order.

have been produced on a commercial basis. The proposed LSI can produce efficient stream files from binary image files that can easily be handled by the new EB systems, by developing simple software that converts the box format, produced by the proposed LSI, to a conventional stream format.

6. Summary

Based on nature-inspired computing, we proposed a CA LSI architecture that extracted quadrilateral objects from binary images. We proposed a serial combination of parallel CA algorithms, based on the RD chemical systems model. Each cell in the CA was implemented by a simple digital circuit called an elemental processor. The CA LSI can be constructed by a large number of elemental processors and their controllers operating in serial and parallel. When we assumed a $0.25\ \mu\text{m}$ CMOS process with the proposed circuits, the estimated total area of the elemental processor was approximately $15\ \text{by}\ 15\ \mu\text{m}^2$.

References

- [1] Adamatzky, A., 2001, *Computing in Nonlinear Media and Automata Collectives* (Bristol: Institute of Physics Publishing).
- [2] Gerhardt, M., Schuster, H. and Tyson, J.J., 1990, A cellular automaton model of excitable media, *Physica D*, **46**, 392–415.
- [3] Young, D., 1984, A Local Activator–Inhibitor Model of Vertebrate Skin Patterns, *Mathematical Biosciences*, **72**, 51.
- [4] Adamatzky, A., 1994, Reaction–diffusion algorithm for constructing discrete generalized Voronoi diagram, *Neural Network World*, **6**, 635–643.
- [5] Adamatzky, A., 1996, Voronoi-like partition of lattice in cellular automata, *Mathematical and Computer Modelling*, **23**, 51–66.
- [6] Adamatzky, A., Arena, P., Basile, A., Carmona-Galán, R., De Lacy Costello, B., Fortuna, L., Frasca, M. and Rodríguez-Vázquez, A., 2004, Reaction–diffusion navigation robot control: From chemical to VLSI analogic processors, *IEEE Transactions on Circuit and Systems I*, **51**(5), 926–938.
- [7] Asai, T., Nishimiya, Y. and Amemiya, Y., 2002, A CMOS reaction–diffusion circuit based on cellular-automaton processing emulating the Belousov–Zhabotinsky reaction, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer*, **E85-A**(9), 2093–2096.
- [8] Bonaiuto, V., Maffucci, A., Miano, G., Salerno, M., Sargeni, F., Serra, P. and Visone, C., 2001, Hardware implementation of a CNN for analog simulation of reaction–diffusion equations. *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems*, Vol. 3, pp. 485–488.
- [9] Matsubara, Y., Asai, T., Hirose, T. and Amemiya, Y., 2004, Reaction–diffusion chip implementing excitable lattices with multiple-valued cellular automata, *IEICE Electronics Express*, **1**(9), 248–252.
- [10] Rekeczky, T., Roska, R., Carmona, F., Jiménez-Garrido, A. and Rodríguez-Vázquez, A., 2003, Exploration of spatial-temporal dynamic phenomena in a 32×32 -cell stored program two-layer CNN universal machine chip prototype, *Journal of Circuits, Systems and Computers*, **12**(6), 691–710.
- [11] Shi, B.E. and Luo, B.T., 2004, Spatial pattern formation via reaction–diffusion dynamics in $32 \times 32 \times 4$ CNN chip, *IEEE Transactions on Circuit and Systems I*, **51**(5), 939–947.
- [12] Asai, T., De Lacy Costello, B. and Adamatzky, A., 2005, Silicon implementation of a chemical reaction–diffusion processor for computation of Voronoi diagram, *International Journal of Bifurcation and Chaos*, **15**(10), in press.
- [13] Asai, T., Adamatzky, A. and Amemiya, Y., 2004, Towards reaction–diffusion computing devices based on minority-carrier transport in semiconductors, *Chaos Solitons & Fractals*, **20**(4), 863–876.
- [14] Asai, T., Kanazawa, Y., Hirose, T. and Amemiya, Y., 2005, Analog reaction–diffusion chip imitating the Belousov–Zhabotinsky reaction with Hardware Oregonator Model, *International Journal of Unconventional Computing*, **1**(2), in press.
- [15] Daikoku, T., Asai, T. and Amemiya, Y., 2002, An analog CMOS circuit implementing Turing’s reaction–diffusion model. *Proceedings of the 2002 International Symposium on Nonlinear Theory and its Applications*, pp. 809–812.

- [16] Karahaliloglu, K. and Balkir, S., 2004, An MOS cell circuit for compact implementation of reaction–diffusion models. *Proceedings of the 2004 International Joint Conference on Neural Networks*, **2004**, p. 1222.
- [17] Serrano-Gotarredona, T. and Linares-Barranco, B., 2003, Log-domain implementation of complex dynamics reaction–diffusion neural networks, *IEEE Transactions on Neural Networks*, **14**(5), 1337–1355.
- [18] Oya, T., Asai, T., Fukui, T. and Amemiya, Y., 2005, Reaction–diffusion systems consisting of single-electron circuits, *International Journal of Unconventional Computing*, **1**(2), in press.
- [19] Field, R.J. and Burger, M., 1985, *Oscillations and Travelling Waves in Chemical Systems* (New York, John Wiley & Sons Inc.).
- [20] Ciletti, M.D., 2002, *Advanced Digital Design with the Verilog HDL* (NJ: Prentice Hall).