# A Quadrilateral-Object Composer for Binary Images with Reaction-Diffusion Cellular Automata

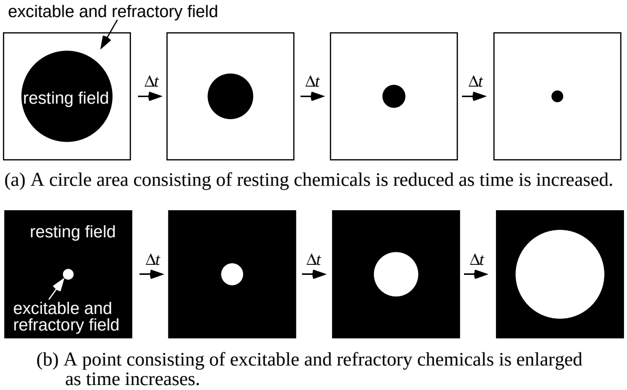
Masayuki Ikebe, Tetsuya Asai, Tetsuya Hirose, and Yoshihito Amemiya
Graduate School of Information Science and Technology, Hokkaido University,
Kita 13, Nishi 8, Kita-Ku, Sapporo 060-8628, Japan.

*Abstract*—**We will describe a cellular-automaton (CA) LSI that extracts quadrilateral objects, such as box areas filled with the same pixel values, from binary images. We propose an efficient CA algorithm, based on the reaction-diffusion chemical systems model. Each cell in the proposed CA is implemented by a digital circuit called an elemental processor. The CA LSI is constructed by a large number of elemental processors operating in parallel. This parallel operation ensures fast and efficient object extraction when the number of cells (processing pixels) increases. When we assumed a 0.25-$\mu$m CMOS process with the proposed circuits, the total area of the elemental processor was approximately 5 by 5 $\mu$m$^2$.**

## I. 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) [2]. Most RD systems can be modeled in terms of CA. For instance, 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. In this paper, we propose a new idea for image processing on chemical processors with complementary object reduction and enlargement operations. Then, we expand this idea into the CA. The proposed CA can detect quadrilateral objects, such as box areas filled with the same pixel values, from binary images. Finally, we propose parallel architecture to implement the proposed CA on VLSIs, and show its performance by numerical simulations.


excitable and refractory field
resting field
Δt
Δt
Δt
resting field
excitable and refractory field
Δt
Δt
Δt

(a) A circle area consisting of resting chemicals is reduced as time is increased.

(b) A point consisting of excitable and refractory chemicals is enlarged as time increases.

Fig. 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).

## II. 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. Generally, three circulative states are introduced according to each unit's dynamical phase; e.g., resting, excitable, and refractory states. In the Belousov-Zhabotinsky (BZ) chemical processor [3], the resting, excitable, and refractory states represent a depletion in the $\mathrm{Br}^-$ ion, an autocatalytic increase in the $\mathrm{HBrO}_2$ ion (oxidation of the catalyzer), and a depletion in the $\mathrm{Br}^-$ ion (reduction of the catalyzer), respectively. 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 the basic ideas of object scaling in a chemical processor. At present, 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

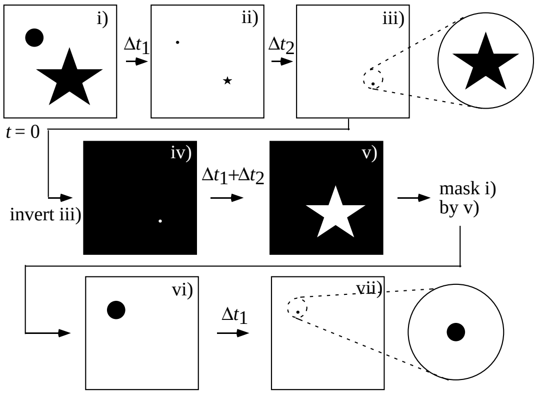
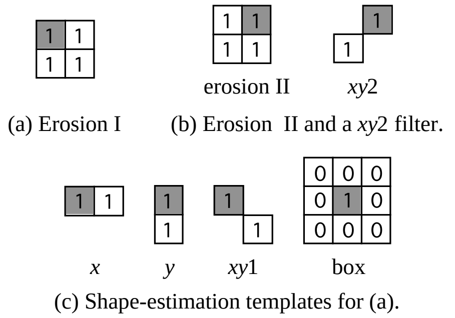


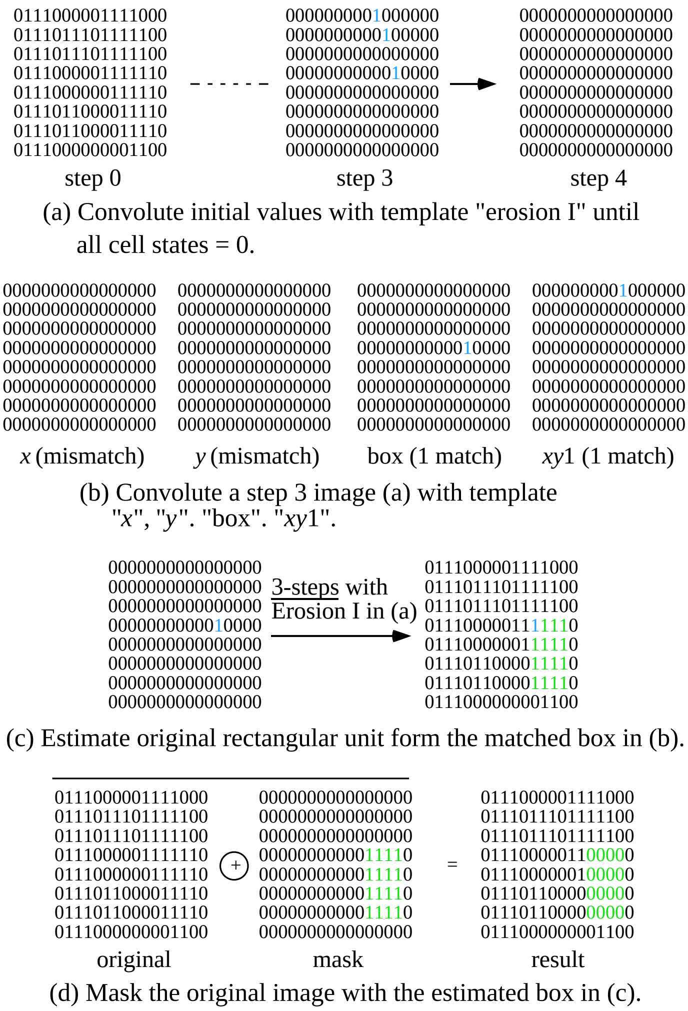
Fig. 2. Object scaling in a chemical processor.



(a) Erosion I (b) Erosion II and a *xy*2 filter.

(c) Shape-estimation templates for (a).

Fig. 3. Templates for proposed CA.



```
0111000001111000             0000000001000000      0000000000000000
0111011101111100             0000000000100000      0000000000000000
0111011101111100             0000000000000000      0000000000000000
0111000001111110             0000000000010000      0000000000000000
0111000000111110  - - - - -  0000000000000000  ->  0000000000000000
0111011000011110             0000000000000000      0000000000000000
0111011000011110             0000000000000000      0000000000000000
0111000000001100             0000000000000000      0000000000000000
     step 0                       step 3                step 4
```

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

```
0000000000000000  0000000000000000  0000000000000000  0000000001000000
0000000000000000  0000000000000000  0000000000000000  0000000000000000
0000000000000000  0000000000000000  0000000000000000  0000000000000000
0000000000000000  0000000000000000  0000000000010000  0000000000000000
0000000000000000  0000000000000000  0000000000000000  0000000000000000
0000000000000000  0000000000000000  0000000000000000  0000000000000000
0000000000000000  0000000000000000  0000000000000000  0000000000000000
0000000000000000  0000000000000000  0000000000000000  0000000000000000
 x (mismatch)      y (mismatch)     box (1 match)     xy1 (1 match)
```

(b) Convolute a step 3 image (a) with template "*x*", "*y*". "box". "*xy*1".

```
0000000000000000                     0111000001111000
0000000000000000  3-steps with       0111011101111100
0000000000000000  Erosion I in (a)   0111011101111100
0000000000010000                     0111000001111110
0000000000000000  ---------------->  0111000000111110
0000000000000000                     0111011000011110
0000000000000000                     0111011000011110
0000000000000000                     0111000000001100
```

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

```
0111000001111000     0000000000000000     0111000001111000
0111011101111100     0000000000000000     0111011101111100
0111011101111100     0000000000000000     0111011101111100
0111000001111110  ⊕  0000000000011110  =  0111000001100000
0111000000111110     0000000000011110     0111000000100000
0111011000011110     0000000000011110     0111011000000000
0111011000011110     0000000000011110     0111011000000000
0111000000001100     0000000000000000     0111000000001100
    original               mask                result
```

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

Fig. 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.

units that exist along the boundary between the object and background, as shown in Fig. 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 Fig. 1(b).

Let us consider the transient shapes of objects during erosion, when the initial input i) in Fig. 2 is given to a chemical processor. The input includes two different objects (a circle and star), where each object is defined by a black (resting) area. As the time increases, the size of each object is reduced due to erosion. Here, we can assume that the circular object is about to disappear at $t = \Delta t_1$ [ii) in Fig. 2], while the star object is about to disappear at at $t = \Delta t_1 + \Delta t_2$ [iii) in Fig. 2]. At iii), the shape of the star is left unchanged, as long as the resolution of the unit array is higher than the display resolution. This makes the star the smallest and most efficient representation of an object in the chemical computer. Consequently, we can consider the chemical processor to be a video memory that has infinite (but molecular level) resolution. To restore the original object, we can simply invert image iii) in Fig. 2, which results in iii). At this point, the chemical computer automatically performs dilation. When $\Delta t_1 + \Delta t_2$ have passed since the start of dilation, the star object is restored in the processor to its original size [v) as shown in Fig. 2]. To memorize the resting (circle) object, we perform further erosions with the initial inputs vi) obtained from the mask of i) by v). When $\Delta t_1$ 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)].

## III. Cellular Automata for Quadrilateral-Object Extraction

Based on the memory model of chemical computers, we propose a CA model that is suitable for VLSI implementation. To imitate the natural erosion described in the preceding section, we used the "erosion I" template, as shown in Fig. 3(a). Figure 4(a) shows an example of erosion on a conventional CA array using this template. In this example, two objects (both represented by clusters of "1") are about to disappear at step 3. Subsequently, we estimate the smallest object type in the reduced image (step 3) by using four templates; i.e., "$x$", "$y$", "$xy1$" and a "box", as shown in Fig. 3(c). In this example, the "$xy1$" and "box" templates can be matched [Fig. 4(b)]. If no templates can be matched, the smallest object must match template "$xy2$", as seen in Fig. 3(b). In this case, we have to restart the erosion process using the"erosion II" template shown in Fig. 3(b).

Once the smallest object type is determined, we can es-

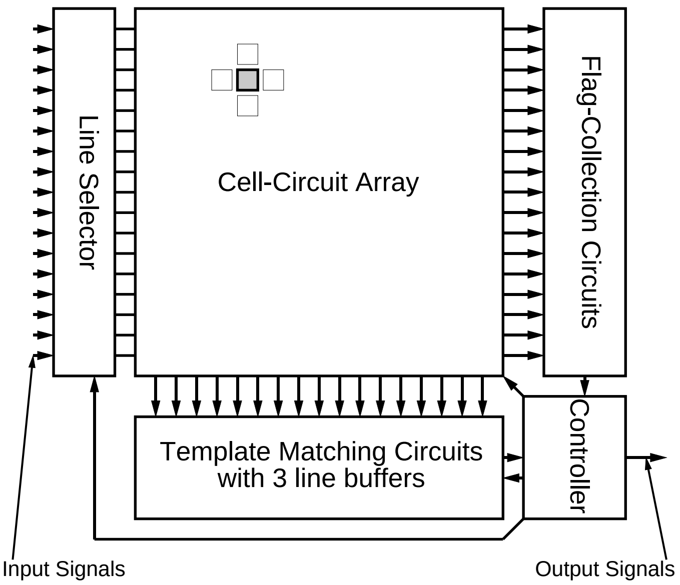


Fig. 5. Block diagram of a quadrilateral object composer for binary images with reaction-diffusion cellular automata.

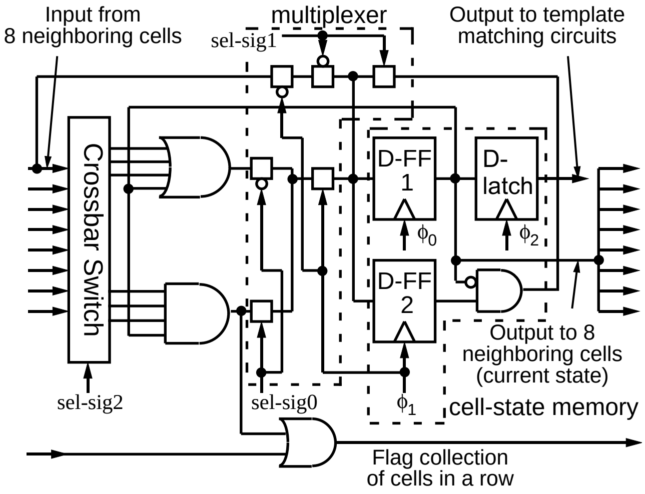

Input from 8 neighboring cells
multiplexer
sel-sig1
Output to template matching circuits
Crossbar Switch
D-FF 1
D-latch
$\phi_0$
$\phi_2$
D-FF 2
$\phi_1$
Output to 8 neighboring cells (current state)
cell-state memory
sel-sig2
sel-sig0
Flag collection of cells in a row

Fig. 6. Cell circuit with four sub-circuits.

timate the maximum "box" area that can be stored within the original object, from the number of steps in the applied template, as shown in Fig. 4(c). At this point, we must memorize the object's box information. Notice that this box is the largest box in the initial image [step 0 in Fig. 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 Fig. 4(d).

## IV. System Implementation

Figure 5 shows a block diagram of the whole 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 ($\phi_1$, $\phi_2$, sel-sig0 → 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", $\phi_0$ supplies a CLK signal that starts from the logical "0", and $\phi_1$ 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 Fig. 5 , the memory circuits of all the cells can obtain the initial data sequentially. After this operation, when $\phi_0$ and $\phi_1$ are set to logical the "1", the initial data can be stored. When sel-sig0 is set to "0" ( or "1"), and $\phi_0$ receives a CLK signal, the cell circuit performs erosion (or dilation) on the CA cell. To obtain an instant when the cell state are all "0", the controller recognizes each subsequent cell state as a flag at each time step. By setting $\phi_0$ to "0", the current cell state is stored in the D-latch for output to the template circuits. We use sel-sig1 and $\phi_1$ 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 can not extract any objects by repeating the first to fourth periods, the fifth period is processed once and the above four-periods sequence is repeated. If the system still can not extract any objects, the system operations are

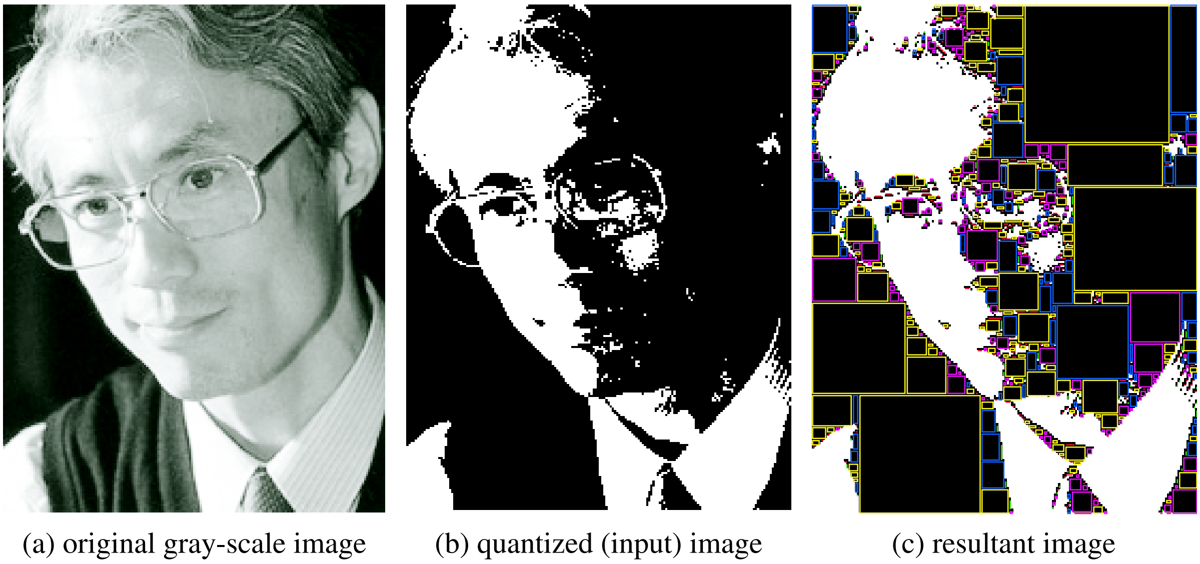


Fig. 7. Simulation results for 180 × 240 image; (a) original image, (b) quantized image given to CA LSI, (c) mixed image of detected quadrilateral objects and (b). The maximum boxes are detected in order.

completed.

## V. Simulation Examples

We conducted numerical simulations for a 180 × 240 pixel image. A natural image [Fig. 7(a)] was quantized, as shown in Fig. 7(b), and input to the proposed CA LSI. The simulation flow is as follows:

1. Transfer the input binary image [Fig. 7(b)] 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 become zero ($\equiv \mathrm{s}_c$) 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 $\mathrm{box}_c$) 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 the processes is counted, and the number at which all the cell values become zero ($\equiv \mathrm{is}_c$) 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 $x_c$, $y_c$, $xy1_t$ and $\mathrm{box}_c$, and create subsequent binary input images, masked by this quadrilateral box.
5. Repeat tasks 1 $\sim$ 4 until $\mathrm{s}_c$ or $\mathrm{is}_c$ 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 $\mathrm{box}_c$.

Figure 7(c) show the results. The maximum boxes were correctly detected in order, as predicted. Notice that this output contains only rectangles and squares, since slanted objects were removed during the last process (6).

## VI. Summary

We propose a cellular-automaton (CA) LSI that extracts quadrilateral objects from binary images. The LSI was constructed by a large array of elemental processors, operating in parallel, which ensured fast and efficient object extraction. When we assumed a 0.25-$\mu$m CMOS process with the proposed circuit, the total area of the elemental processor was approximately 5 by 5 $\mu\mathrm{m}^2$.

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 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.

## References

[1] A. Adamatzky, *Computing in Nonlinear Media and Automata Collectives.* Institute of Physics Publishing, Bristol, 2001.

[2] T. Toffoli and N. Margolus, *Cellular Automata Machines.* MIT Press, Cambridge, 1987.

[3] R. J. Field and M. Burger, *Oscillations and travelling waves in chemical systems.* John Wiley & Sons, Inc., 1985.