Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA, July 31 – August 5, 2011

Cellular Neural Networks with Switching Two Types of Templates

Yoshihiro Kato, Yasuhiro Ueda, Yoko Uwate and Yoshifumi Nishio

***Abstract*— In this study, we propose Cellular Neural Networks with switching two types of templates. In the CNN, space varying system is known that it can perform complex processing. Generally, the space varying CNN is not easy to design. However, we can set existing template on each cell of CNN by the proposed method. In binarization, complex portions of input image are not processed well by using the conventional CNN. On the other hand, the complex portion can be processed well by the proposed method. In the edge detection, the indistinct portion is not detected by the conventional CNN with "$Edge\ detection$" template of 3×3 matrix. It is difficult for CNN to recognize that it is the edge or not. Additionally, the detected edge is too bold and some noises are left with "$Edge\ detection$" template of 5×5 matrix. By switching these templates in case, we can detect edge in indistinct position. In pattern formation, generally, simple pattern is formed by using one template. On the other hand, some complex patterns are formed by the proposed method. From some simulation results, we confirm that the proposed method is effective for various image processing.**

I. Introduction

In 1988, Cellular Neural Networks (CNN) [1] were introduced by L. O. Chua and L. Yang. The idea of the CNN was inspired from the architecture of the cellular automaton and the neural networks. A different point from the conventional neural networks is that CNN has local connectivity property. Since, the structure of CNN resembles that of animals' retina. Therefore, CNN can be used for various image processing applications [2]-[5] including pattern formation [6]. CNN is composed of the basic analog circuit units called cells. The cell contains linear and nonlinear circuit elements which are typically linear capacitors, linear resistors, linear and nonlinear controlled sources. The performance of CNN depends on the parameters called the template. The template of CNN consists three elements as the feedback template A, the control template B, and the constant bias I.

Generally, the conventional CNN depends on an input image because of space invariant template. In the binarization, the complex portions of input image are not processed well. Also, the processed image is rough by using only "$Smoothing$" template. In edge detection, the edge of the indistinct portion is not detected by using "$Edge\ detection$" template of 3×3 matrix. On the other hand, by using "$Edge\ detection$" template of 5×5 matrix, we can detect the edge of the indistinct portion. However, the detected edge becomes bold line and some noises are left. In pattern formation, formed pattern is simple pattern by using one template, like ripple and checkerboard pattern. Therefore, we note space varying CNN which can perform processing more complex than the conventional CNN. However, space varying CNN is not easy to design.

In this study, we propose CNN with switching two types of templates. Applied templates are decided by the input, the output value, and the boundary parameters which are D_1 and D_2. In the proposed method, we decide D_1 and D_2 as step 1. Then, the template is determined corresponding to the input value of each cell, D_1 and D_2. As step 2, the CNN processes the input image by the applied template. As step 3, the steps 1 and 2 are carried out by using the output value. As step 4, steps 2 and 3 are repeated. The switching is performed according to time and the value of cell. The proposed method is space and time varying. We show some simulation results and confirm its effectiveness.

The rest of this paper is structured as follows. In Sec. 2, we review the basic of the standard CNN. In Sec. 3, we show the algorithm of the CNN with switching two types of templates. In Sec. 4, some simulation results of the proposed method are shown. The section 5 concludes the article.

II. Cellular Neural Networks

In this section, we explain the basic structure of the CNN. The CNN has M by N processing unit circuits called cells. The cell contains linear and nonlinear circuit elements which are typically linear capacitors, linear resistors, linear and nonlinear controlled sources. Cells are arranged in a reticular pattern to M line N row. We represent a cell $C(i,j)$ using a variable i which denotes vertical position and a variable j which denotes horizontal position. The CNN is an array of cells. Each cell is connected to its neighboring cells according to a template. Usually, the template is the same for all cells except for boundary cells. The CNN has the features of time continuity, spatial discreteness, nonlinearity and parallel processing capability.

State equation :

$$\begin{aligned}\frac{dv_{xij}}{dt} &= -v_{xij} + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} A_{(i,j;k,l)}v_{ykl}(t) \\ &\quad + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} B_{(i,j;k,l)}v_{ukl}(t) + I, \qquad (1)\end{aligned}$$

Yoshihiro Kato, Yasuhiro Ueda, Yoko Uwate and Yoshifumi Nishio are with the Department of Electrical and Electronic Engineering, Tokushima University, 2-1 Minami-Josanjima, Tokushima 770-8506, Japan (Phone: +81 88 656 7470; email: {kkato, yasuhiro, uwate, nishio}@ee.tokushima-u.ac.jp).

978-1-4244-9636-5/11/$26.00 ©2011 IEEE

Output equation :

$$v_{yij}(t) \;=\; \frac{1}{2}(|v_{xij}(t)\;+\;1|\;-\;|v_{xij}(t)\;-\;1|), \quad (2)$$

where v_x, v_y and v_u represent a state, an output and an input of cell, respectively. In the Eq. (1), A is the feedback template and B is the control template. These and the constant bias I are collectively called general template. The output equation is a piece-wise linear function. In fact, The output value of CNN is within of -1 to 1.

The $r-neighborhood$ of $C(i,j)$ in CNN is defined by

$$\begin{aligned} Nr(i,j) \;=\; \{C(k,l) \mid max\{\, |k-i|,\ |l-j|\ \} \le r, \\ 1 \le k \le M; 1 \le l \le N\}, \quad (3) \end{aligned}$$

where r is a positive integer number. Each cell has $(2r+1)^2$ neighborhood cells. Hence, the size of template is defined by r. If r is 1, the size of template is 3×3 matrix. Additionally, if r is 2, the size of template is 5×5 matrix.

III. CNN with Switching Template

In this section, we explain the algorithm of the proposed method. The proposed method is that two types of templates are switched corresponding to the input and the output values. The algorithm is shown as follows.

Step 1: We decide the boundary parameters which are D_1 and D_2 ($D_1{<}D_2$). By these values, applied templates are decided.

In the switching rule, the template is determined corresponding to the value of cell. If v_u is under D_1 or over D_2, the "Template 1" is applied to the cell. While, if input value v_u is over D_1 and under D_2, the "Template 2" is applied like Fig. 1.

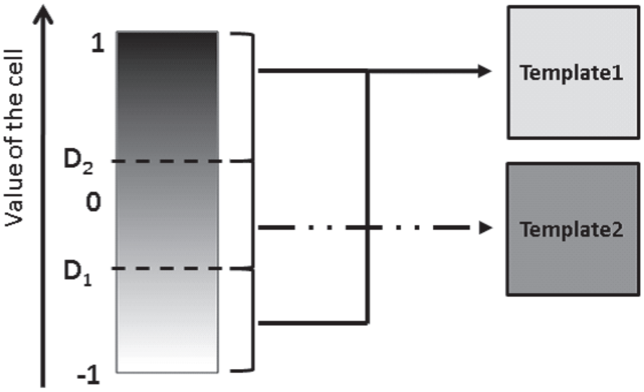


Fig. 1. Switching rule.

Step 2: The value of each cell is updated by the state equation and the output equation as follows.

State equation with the "Template 1" :

$$\begin{aligned} \frac{dv_{xij}}{dt} \;&=\; -v_{xij} + \sum_{k=i-r_1}^{i+r_1}\sum_{l=j-r_1}^{j+r_1} A_1(i,j;k,l)v_{xkl}(t) \\ &+ \sum_{k=i-r_1}^{i+r_1}\sum_{l=j-r_1}^{j+r_1} B_1(i,j;k,l)v_{ukl}(t) + I_1 \\ &(|i-k| \le r_1\ ,\ |j-l| \le r_1). \quad (4) \end{aligned}$$

State equation with the "Template 2" :

$$\begin{aligned} \frac{dv_{xij}}{dt} \;&=\; -v_{xij} + \sum_{k=i-r_2}^{i+r_2}\sum_{l=j-r_2}^{j+r_2} A_2(i,j;k,l)v_{xkl}(t) \\ &+ \sum_{k=i-r_2}^{i+r_2}\sum_{l=j-r_2}^{j+r_2} B_2(i,j;k,l)v_{ukl}(t) + I_2 \\ &(|i-k| \le r_2\ ,\ |j-l| \le r_2). \quad (5) \end{aligned}$$

Output equation :

$$v_{yij}(t) = \frac{1}{2}(|v_{xij}(t)+1| - |v_{xij}(t)-1|). \quad (6)$$

Step 3: After 0.005 [τ], the Steps 1 and 2 are carried out by the output value v_y. If v_y is under D_1 or over D_2, the "Template 1" is applied to the cell. While, if v_y is over D_1 and under D_2, the "Template 2" is applied.

Then, the proposed CNN switches the "Template 1" and "Template 2" according to the output value of the cell at every 0.005 [τ].

Step 4: Steps 2 and 3 are repeated every 0.005 [τ].

Therefore, input image is processed by the two types of templates which are determined by the input, the output and the boundary parameters.

IV. Simulation Results

In this section, we show some simulation results for binarization, edge detection and pattern formation by using the proposed method. The used templates are found in [7] and [8].

A. Binarization

In this subsection, we show some simulation results of binarization by using the proposed method. For this simulation, we use some general images including complex portion. To emphasize complex portion, we use "$Edge\ detection$" template of 3 × 3 matrix. "Template 1" and "Template 2" are set to "$Smoothing$" template and "$Edge\ detection$" template of 3 × 3 matrix, respectively. These templates are same size and carry out different processing. Also, r_1 and r_2 are set to 1. Templates for binarization are shown as follows.

Smoothing template :
This template can binarize of image.

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ I = 0. \quad (7)$$

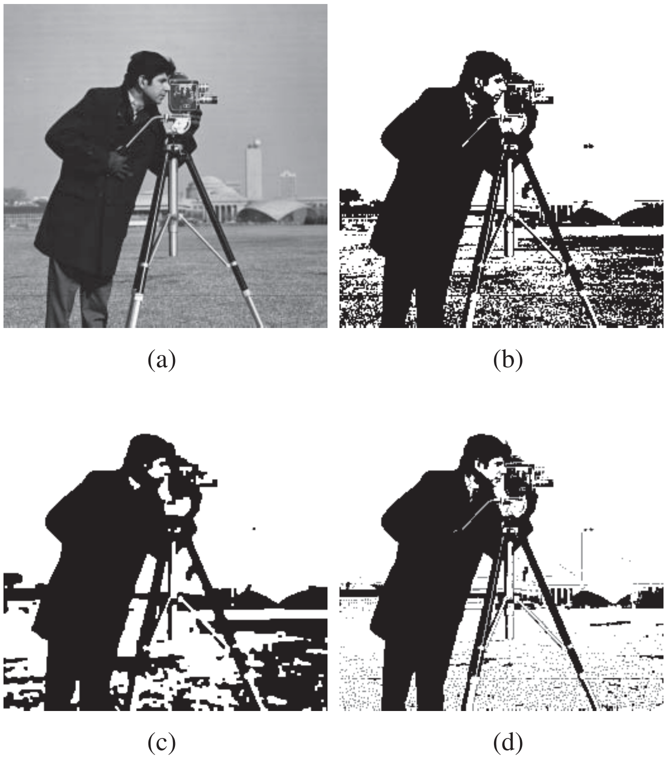


Fig. 2. Simulation results 1. (a) Input image (Cameraman). (b) Binarization result by the digital processing. (c) Binarization result using the conventional CNN. (d) Binarization result using the proposed method. (D_1=−0.2, D_2=0.2).

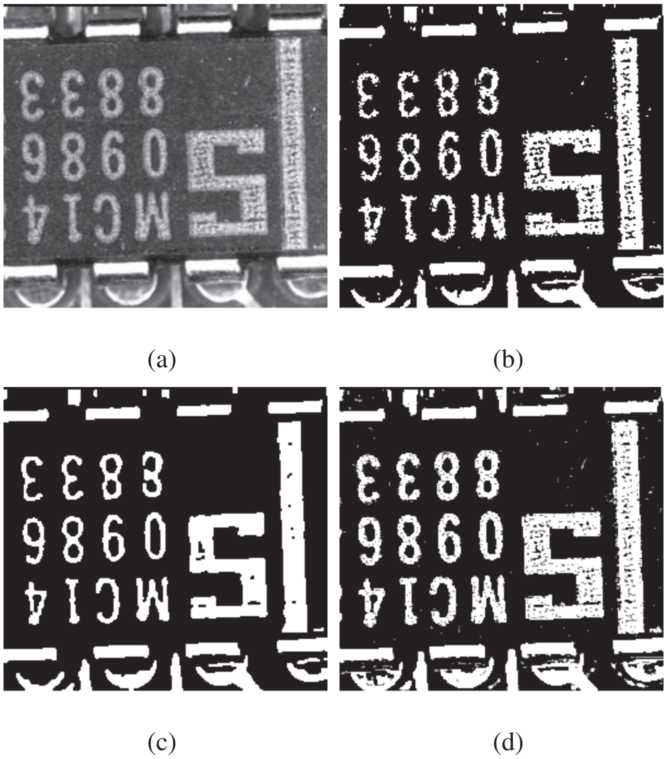


Fig. 3. Simulation results 2. (a) Input image (Text). (b) Binarization result using the digital processing. (c) Binarization result using the conventional CNN. (d) Binarization result using the proposed method. (D_1=−0.1, D_2=0.2).

Edge detection template of 3×3 matrix :
This template can detect the edge of objects.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, I = -1. \tag{8}$$

In this simulation, output images were compared with the images which are binarized by digital processing. The binarization method of digital processing are shown as follows. If the value of each pixel is over zero, the pixel turn to black. On the other hand, if the value of each pixel is under zero, the pixel turn to white.

Figure 2 shows the simulation results using the proposed method. In Fig. 2 (a), this input image including complex portions such a camera and the pillars of buildings in the background. D_1 and D_2 are set to -0.2 and 0.2. Figure. 2 (b) compares with binarized output image by the digital processing. In Fig. 2 (c), the complex portions are not processed well by the conventional CNN. In Fig. 2 (d), binarization of the complex portion can be processed well by using the proposed method.

Figure 3 shows the simulation results for another input image. In Fig. 3 (a), this input image including complex portions such a some texts and lower of image. D_1 and D_2 are set to -0.1 and 0.2. Figure 3 (b) compares with binarized output image by the digital processing. In Fig. 3 (c), binarization of complicated portions cannot be processed well by the conventional CNN. On the other hand, in Fig. 3 (d), binarization of the complex portions which like cracks on board and uneven paints can be processed well by using the proposed method.

From these results, we consider that the proposed method is more effective than the conventional CNN for binarization.

B. Edge detection

In this subsection, we show some simulation results of edge detection using the proposed method. For this simulation, we use some general images including indistinct portion. In these images, the value of the center cell is close to that values of neighborhoods in many cases. There are included indistinct portion especially. Hence, the edge of the indistinct portion is not detected by using "$Edge\ detection$" template of 3×3 matrix. To feed off information on the two-neighborhood value of cell, we extend the "$Edge\ detection$" template like Eq. (9). In fact, "Template 1" and "Template 2" are set to "$Edge\ detection$" template of 3×3 matrix and "$Edge\ detection$" template of 5×5 matrix, respectively. These templates are different size and carry out similar processing. Also, r_1 and r_2 are set to 1 and 2, respectively.

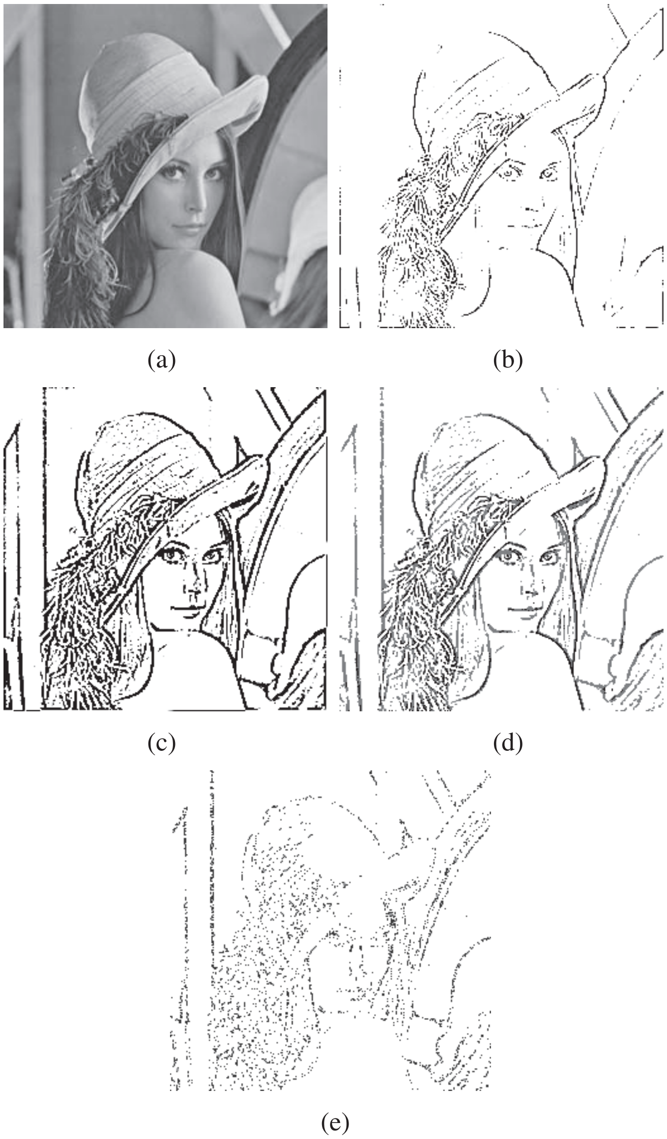


Fig. 4. Simulation results 3. (a) Input image (Lenna). (b) Edge detection result using the 3×3 matrix template. (c) Edge detection result using the 5×5 matrix template. (d) Edge detection result using the proposed method. (D_1=−0.2, D_2=0.2). (e) Distribution of 3×3 and 5×5 matrix template.

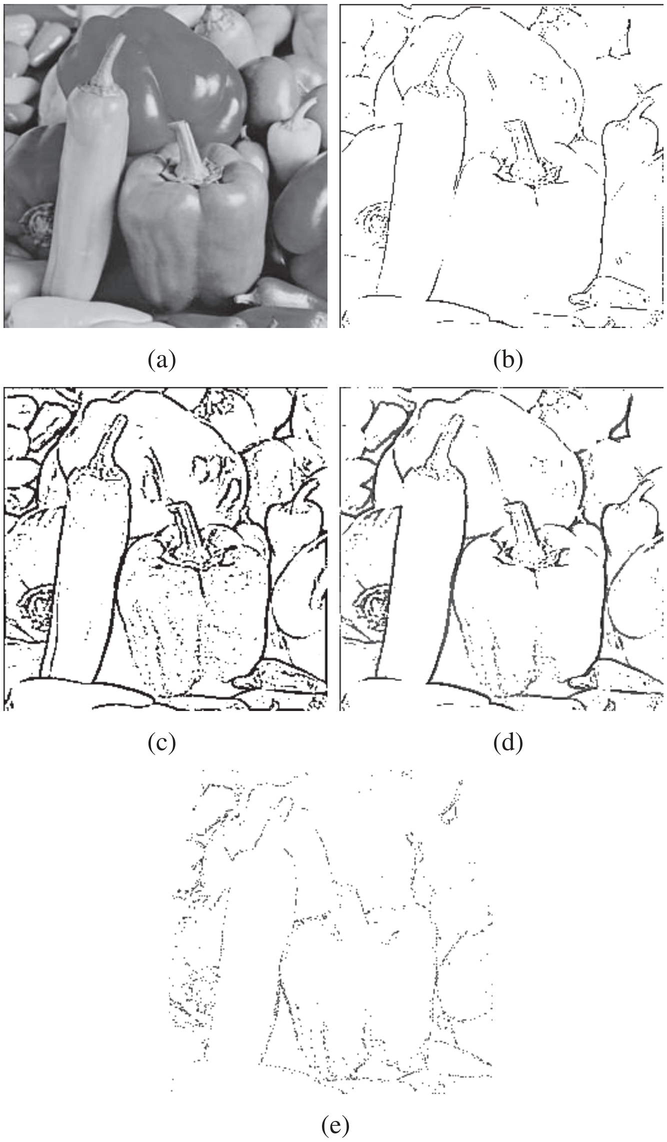


Fig. 5. Simulation results 4. (a) Input image (Pepper). (b) Edge detection result using the 3×3 matrix template. (c) Edge detection result using the 5×5 matrix template. (d) Edge detection result using the proposed method. (D_1=0.3, D_2=0.6). (e) Distribution of 3×3 matrix template and 5×5 matrix template.

Edge detection template of 5×5 matrix :

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ B &= \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 16 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}, \\ I &= -1. \end{aligned} \tag{9}$$

Figure 4 shows the simulation results using the proposed method. In Fig. 4 (a), D_1 and D_2 are set to -0.2 and 0.2. In Fig. 4 (a), the pillar on the left side in the background is an indistinct portion. In Fig. 4 (b), this indistinct portion is not detected using the conventional CNN with the "*Edge detection*" template of 3×3 matrix. In Fig. 4 (c), although the edge of the indistinct portion is detected using the conventional CNN with the "*Edge detection*" template of 5×5 matrix, the detected edge becomes bold line and some noises are left. In Fig. 4 (d), using the proposed method, the edge of the indistinct portion can be detected more clearly and the detected edge is more fine line than using the conventional CNN. In Fig. 4 (e), black points show the cell using the "*Edge detection*" template of 3×3 matrix and white points show the cell using the "*Edge detection*" template of 5×5 matrix. From Fig. 4 (e), we can refer that distinct parts are processed by the "$Edge\ detection$" template of 3×3 matrix and indistinct parts are processed by the "$Edge\ detection$" template of 5×5 matrix. In fact, the part of edge which is difficult to detect is processed by the "$Edge\ detection$" template of 5×5 matrix.

Figure 5 shows the simulation results for another input image. In Fig. 5 (a), D_1 and D_2 are set to 0.3 and 0.6. The almost edge is not detected using the conventional CNN with the "$Edge\ detection$" template of 3×3 matrix in Fig. 5 (b). On the other hand, the detected edge becomes bold line and some noises are left using the conventional CNN with the "$Edge\ detection$" template of 5×5 matrix in Fig. 5 (c). In Fig. 5 (d), by using the proposed method, the edge of indistinct potion can be detected more clearly and the detected edge is more fine line than the conventional CNN. Also, compare to Fig. 5 (c), some noises are removed by using the proposed method. Also, the part of edge which is difficult to detect is processed by the "$Edge\ detection$" template of 5×5 matrix.

From these results, we consider that the proposed method is also more effective than the conventional CNN for the edge detection.

C. Pattern formation

In this subsection, we show some simulation results of pattern formation by using the proposed method. To extend pattern area, we use "$Black\ propagation$" template. "Template 1" and "Template 2" are set to "$Black\ propagation$" template and "$Ripple$" template or "$Checkerboard$" template [8], respectively. These templates are different size and carry out different processing. Also, r_1 and r_2 are set to 1 and 2. Templates for pattern formation are shown as follows.

Black propagation template :
This template can increase black areas.

$$\begin{aligned} A &= \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}, \\ B &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ I &= 3.75. \end{aligned} \tag{10}$$

Ripple template :
This template forms a simple ripple pattern.

$$\begin{aligned} A &= \begin{bmatrix} 0 & -0.2 & -0.2 & -0.2 & 0 \\ -0.2 & 0 & 0.2 & 0 & -0.2 \\ -0.2 & 0.2 & 1.5 & 0.2 & -0.2 \\ -0.2 & 0 & 0.2 & 0 & -0.2 \\ 0 & -0.2 & -0.2 & -0.2 & 0 \end{bmatrix}, \\ B &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ I &= 0. \end{aligned} \tag{11}$$

Checkerboard template :
This template forms the black and white pattern with the cycle of 2 cells.

$$\begin{aligned} A &= \begin{bmatrix} -0.3 & -1 & -3 & -1 & -0.3 \\ -1 & 0.5 & -2 & 0.5 & -1 \\ -3 & -2 & -4 & -2 & -3 \\ -1 & 0.5 & -2 & 0.5 & -1 \\ -0.3 & -1 & -3 & -1 & -0.3 \end{bmatrix}, \\ B &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ I &= 0. \end{aligned} \tag{12}$$

Figure 6 shows the simulation results. "Template 1" and "Template 2" are set to "$Black\ propagation$" template and "$Ripple$" template, respectively. Figure 6 (a) is including some objects of different sizes and shapes. In Fig. 6 (b), formed pattern is simple ripple pattern in spite of some objects of different sizes and shapes by using "$Ripple$" template. However, by using the proposed method like Fig. 6 (c), it is notable that formed patterns are different by some objects of different sizes and shapes. That is because extended areas are different by "$Black\ propagation$" template. Moreover, formed patterns are differed by changing the boundary parameters D_1 and D_2 like Figs. 6 (d) and (e).

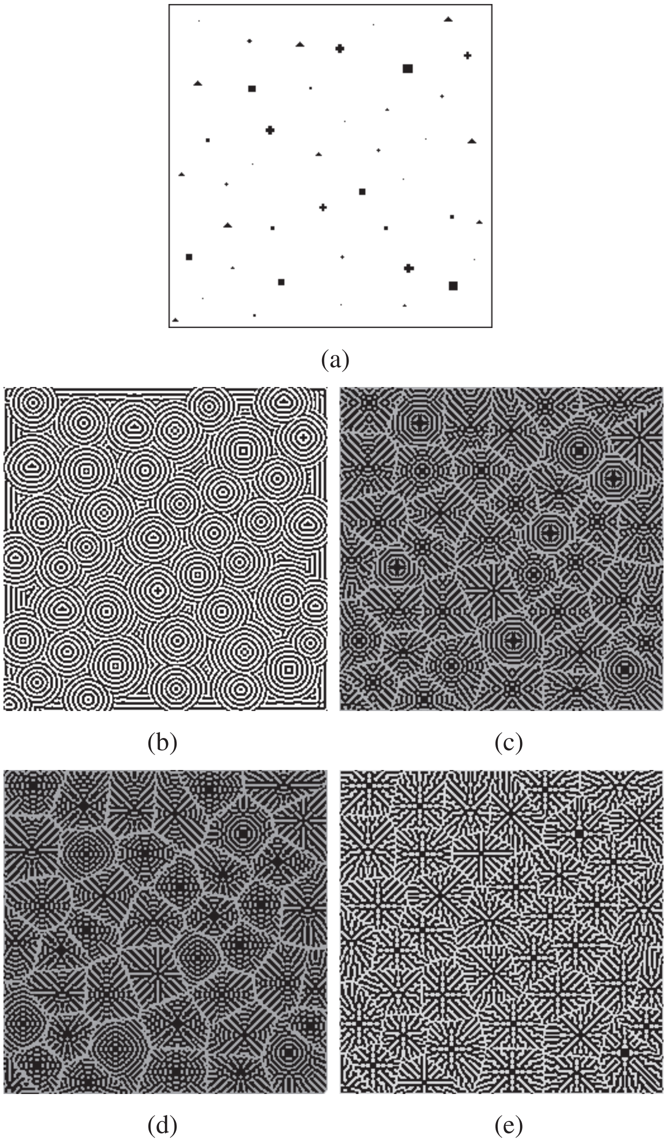
Figure 7 shows the simulation results for another combination of template. "Template 1" and "Template 2" are set to "$Black\ propagation$" template and "$Checkerboard$" template, respectively. In Fig. 7 (a), formed pattern is checkerboard pattern by using "$Checkerboad$" template. However, formed pattern is simple ripple pattern in spite of some objects of different sizes and shapes by using the proposed method like Fig. 7 (c). Moreover, formed patterns are differed by changing the boundary parameters D_1 and D_2 like Figs. 7 (c) and (d). Similar to the above results, formed pattern becomes large, and while the place where formed pattern collided had been held as a boundary line.

From these results, formed pattern changes depending on the each size and shape of objects, the combination of template and the boundary parameters D_1 and D_2 by using the proposed method. Also, more complex patterns are able to be formed .

V. Conclusions

In this study, we have proposed cellular neural networks with switching two types of templates. The proposed method is that two types of templates were switched corresponding to the input and the output values. From simulation results of binarization, the complex portion could be processed better. Furthermore, we could detect the edge in indistinct position. Moreover, from simulation results of pattern formation, we have confirmed that formed pattern changes depending on the each size and shape of objects, the combination of templates and the boundary parameters.

From some simulation results, we consider that our proposed method is effective for various image processing. In



(a)

(b) (c)

(d) (e)

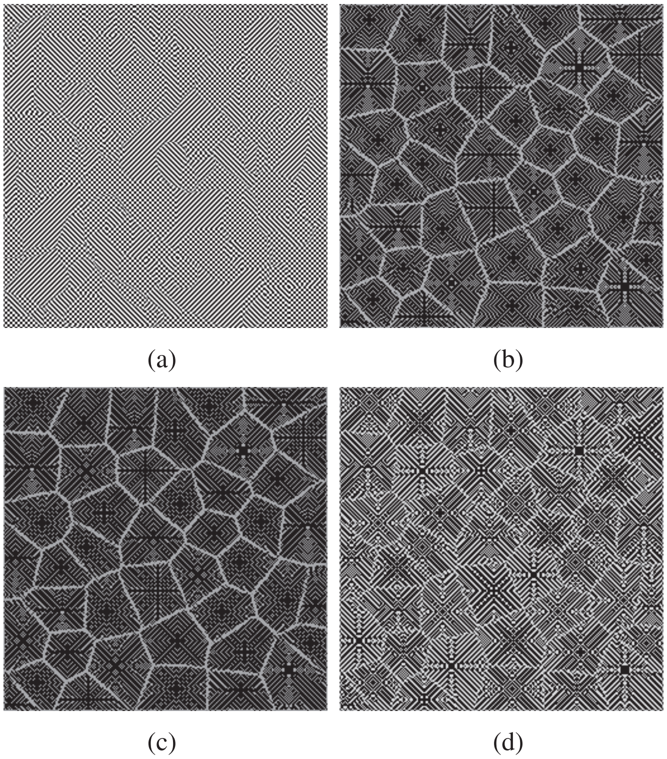
Fig. 6. Simulation results 5. (a) Input image. (b) Pattern formation result by using only "$Ripple$" template. (c) Pattern formation result by switching "$Black\ propagation$" template and "$Ripple$" template. (D_1=−0.3, D_2=0.3). (d) Pattern formation result by switching "$Black\ propagation$" template and "$Ripple$" template. (D_1=−0.2, D_2=0.2). (e) Pattern formation result by switching "$Black\ propagation$" template and "$Ripple$" template. (D_1=−0.7, D_2=0.7).

the future works, we would like to propose that the system automatically decide the best D_1 and D_2 for edge detection and binarization. Also, we investigate the phenomena about pattern formation by the proposed CNN.

References

[1] L. O. Chua and L. Yang, "Cellular Neural Networks: Theory," IEEE Trans. Circuits Syst., vol. 32, pp. 1257-1272, Oct. 1988.

[2] F. Dirk and T. Ronald, "Coding of Binary Image Data using Cellular Neural Networks and Iterative Annealing," Proc. of ECCTD'03, vol. 1, pp. 229-232, Sep. 2003.



(a) (b)

(c) (d)

Fig. 7. Simulation results 6. (a) Pattern formation result by using only "$Checkerboard$" template. (b) Pattern formation result by switching "$Black\ propagation$" template and "$Checkerboard$" template. (D_1=−0.3, D_2=0.3). (c) Pattern formation result by switching "$Black\ propagation$" template and "$Checkerboad$" template. (D_1=−0.2, D_2=0.2). (d) Pattern formation result by switching "$Black\ propagation$" template and "$Checkerboad$" template. (D_1=−0.7, D_2=0.7).

[3] M. Namba and Z. Zhang, "Cellular Neural Network for Associative Memory and Its Application to Braille Image Recognition," Proc. of IJCNN'06, pp. 4716-4721, Jul. 2006.

[4] H. Koeppl and L. O. Chua, "An Adaptive Cellular Nonlinear Network and its Application," Proc. of NOLTA'07, pp. 15-18, Sep. 2007.

[5] T. Kozek, K. R. Crounse, T. Roska and L. O. Chua, "Smart Image Scanning Algorithms for the CNN Universal Machine," Proc. of NOLTA'95, vol. 2, pp. 707-712, 1995.

[6] K. R. Crounse and L. O. Chua, "Methods for Image Processing and Pattern Formation in Cellular Neural Networks: A Tutorial," IEEE Trans. Circuits Syst., vol. 42, no. 10, pp. 583-601, Oct. 1995.

[7] Cellular Sensory Wave Computers Laboratory Computer and Automation Research Institute Hungarian Academy of Sciences, "Cellular wave Computing Library (Template, Algorithms, and Programs) Version 2.1"

[8] T. Nishio and Y. Nishio, "Periodic Pattern Formation and its Applications in Cellular Neural Networks" IEEE Transactions on Circuits and Systems I, vol. 55, no. 9, pp. 2736-2742, Oct. 2008.