IEICE TRANS. FUNDAMENTALS, VOL.E85–A, NO.9 SEPTEMBER 2002

PAPER *Special Section on Nonlinear Theory and Its Applications*

Image Processing of Two-Layer CNNs——Applications and Their Stability——

Zonghuang YANG[†a)], ***Student Member***, **Yoshifumi NISHIO**[††], ***and*** **Akio USHIDA**[††], ***Regular Members***

SUMMARY Cellular Neural Networks (CNNs) have been developed as a high-speed parallel signal-processing platform. In this paper, a generalized two-layer cellular neural network model is proposed for image processing, in which two templates are introduced between the two layers. We found from the simulations that the two-layer CNNs efficiently behave compared to the single-layer CNNs for the many applications of image processing. For examples, simulation problems such as linearly non-separable task—logic XOR, center point detection and object separation, etc. can be efficiently solved with the two-layer CNNs. The stability problems of the two-layer CNNs with symmetric and/or special coupling templates are also discussed based on the Lyapunov function technique. Its equilibrium points are found from the trajectories in a phase plane, whose results agree with those from simulations.

key words: *cellular neural networks, image processing, cloning template, stability*

1. Introduction

Since Cellular Neural Networks (CNNs) were proposed by Chua and Yang [1], [2], they have been successfully developed for various high-speed parallel signal processings such as image processing [3]–[10] as well as modeling of nonlinear phenomena [16]–[21].

Up to the present, many kinds of templates for single-layer and two-layer CNNs have been already proposed for special purposes. Reference [15] provides many templates and algorithms for the applications of image processing with single-layer CNNs. The image processing such as center point detection, skeletonizing and object separation etc. with single-layer CNNs, can be carried out by the iterative use of different time-invariant templates [11]–[14], where each single-layer CNN is iteratively used to perform a part of the task. After the operation of CNN has attained to the steady state or reached at some state, the next single-layer CNN begins to perform the next part of the task. In this way, the process iteratively continues until the whole task is completed. This procedure is really tedious serial image processing. On the other hand, it is well known in the neural networks that if a multi-layer structure neural network is taken into consideration, it will have more wide applications in different fields such as linearly non-separable problem, optimization and pattern recognition, etc. [32]. This mechanism will be also valid to the cellular neural networks. Actually, the concept and the interesting applications of the two and more layer CNN structures have already been proposed in [1], [16]–[21], [27]–[29], where the CNNs are used in generating nonlinear phenomena such as autowaves, pattern formation, Radon transform of a binary image and so on. Another interesting application of two-layer CNN in [30] was proposed as highly parallel method for mapping and navigation of an autonomous robot.

In this paper, we will systematically investigate the ability of CNNs with two-layer structure in image processing. Through our many experimental examples, we found that the two-layer CNNs efficiently behave compared to the single-layer CNNs for many image processing applications.

This paper is organized as follows: In Sect. 2, a two-layer cellular neural network model is described as the extension of the single-layer CNN [1], in which two coupling templates are introduced. In Sect. 3, several interesting simulations for image processing are given in detail. All of them show that the two-layer CNNs are more efficient for the applications of image processing compared to single-layer CNNs. In Sect. 4, the stability of the two-layer CNN with symmetric and/or special coupling templates is discussed based on the Lyapunov function technique, whose equilibrium points are calculated by the trajectories in the phase plane.

2. Two-Layer CNN Architecture

Now, let us formulate the system equations of the two-layer CNNs by introducing two coupling templates C_1 and C_2. We assume that each layer of the CNN is composed of a two-dimensional M by N array structure as shown in the left hand side of Fig. 1. Each cell in the array is denoted by $c(i, j)$, and has two state variables $x_1(i, j)$, $x_2(i, j)$, where (i, j) stands for the position of a cell in the array, for $1 \leq i \leq M$ and $1 \leq j \leq N$. The state equations of each cell are given by two first-

Manuscript received November 5, 2001.
Manuscript revised March 25, 2002.
Final manuscript received May 24, 2002.

[†]The author is with the Graduate School of Engineering, University of Tokushima, Tokushima-shi, 770-8506 Japan.
[††]The authors are with the Department of Electrical and Electronic Engineering, Universuty of Tokushima, Tokushima-shi, 770-8506 Japan.
a) E-mail: yangzh@ee.tokushima-u.ac.jp

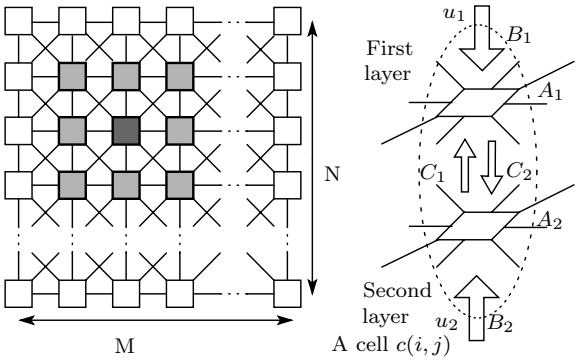


Fig. 1 A two-dimensional cellular neural network, and coupling between the first and second layers.

order differential equations given by Eq. (1), and the output equations are given by Eq. (2), where $f(\cdot)$ is a piecewise-linear nonlinear function defined by Eq. (3). Figure 2 shows the characteristics of the output function. We define the state variables of the first layer by $x_1(i,j)$, and those of the second layer by $x_2(i,j)$. u and y refer to the input and output variables of the cell. $A(i,j;k,l)$, $B(i,j;k,l)$, and I mean the feedback template, filter template and bias current, respectively. The index 1 and 2 stand for the first layer and second layer of the two-layer CNN array. $C_1(i,j;k,l)$ is the coupling template to transfer the second layer output to the first layer input, and $C_2(i,j;k,l)$ is vice versa. They also show the weight of the local couplings among the cells in the neighborhood $N_r(i,j)$ as the templates $A(i,j;k,l)$ and $B(i,j;k,l)$. r is the coupling radius. In general, r takes the value of 1 or 2. An example of $r=1$ for a particular cell is shown by the gray cells in Fig. 1.

$$\left.\begin{aligned}\frac{dx_{1,ij}}{dt} &= -x_{1,ij} + I_1 \\ &\quad + \textstyle\sum_{C(k,l)\in N_r(i,j)} A_1(i,j;k,l)y_{1,kl} \\ &\quad + \textstyle\sum_{C(k,l)\in N_r(i,j)} B_1(i,j;k,l)u_{1,kl} \\ &\quad + \textstyle\sum_{C(k,l)\in N_r(i,j)} C_1(i,j;k,l)y_{2,kl} \\ \frac{dx_{2,ij}}{dt} &= -x_{2,ij} + I_2 \\ &\quad + \textstyle\sum_{C(k,l)\in N_r(i,j)} A_2(i,j;k,l)y_{2,kl} \\ &\quad + \textstyle\sum_{C(k,l)\in N_r(i,j)} B_2(i,j;k,l)u_{2,kl} \\ &\quad + \textstyle\sum_{C(k,l)\in N_r(i,j)} C_2(i,j;k,l)y_{1,kl}\end{aligned}\right\} \quad (1)$$

$$\left.\begin{aligned}y_{1,ij} &= f(x_{1,ij}) \\ y_{2,ij} &= f(x_{2,ij})\end{aligned}\right\} \quad (2)$$

$$f(x) = 0.5(|x+1| - |x-1|). \quad (3)$$

The block diagram of two-layer CNN is shown by Fig. 3, which constitutes a closed-loop system. Observe that they share out the tasks in image processing and cooperate with each other, which cannot be performed with single-layer CNNs. Note that CNNs proposed in some literatures [11]–[15] are serial image processing with single-layer CNNs, and our two-layer CNNs on the contrary are parallel image processing. Moreover, when one of the templates C_1 and C_2 is set to zero, the CNN becomes an open-loop system and behaves as the cascade connection of two single-layer CNNs. Therefore, the two-layer CNNs will have wide applications in image processing compared to the single-layer CNNs.

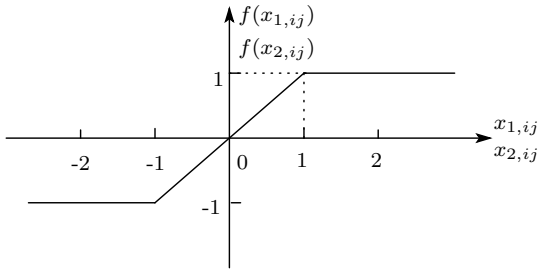


Fig. 2 The CNN output function described by the piecewise-linear sigmoid characteristic.

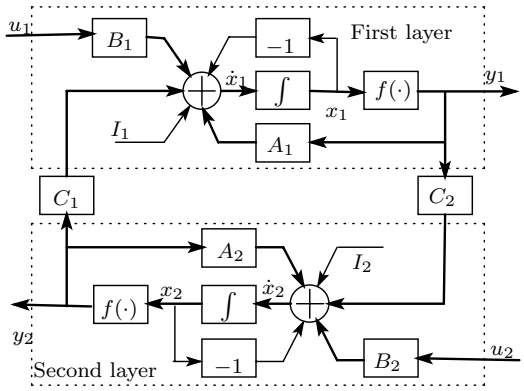


Fig. 3 Block diagram of the two layer CNN.

3. Applications in Image Processing

Image processing is one of the most important applications of the CNNs, and there have been already proposed many kinds of templates for the single-layer CNNs. In this section, we will give some important examples of the two-layer CNNs.

3.1 Linearly Non-separable Problems

In binary value image processing, some logic operations between two images are often introduced such as Logic OR, AND and NOT, etc. The Logic OR, AND and NOT belong to linearly separable problems in image processing, whose templates with single-layer CNNs have been reported in the literature [15]. In fact, this kind of CNNs is extremely useful to realize these logic operations. For some linearly non-separable tasks (e.g. Logic XOR), they cannot be directly solved by the single-layer CNNs. However, we can solve them with the two-layer CNNs, directly.

Now, let us consider a simplified two-layer CNN, i.e., all parameters besides the center positions in the

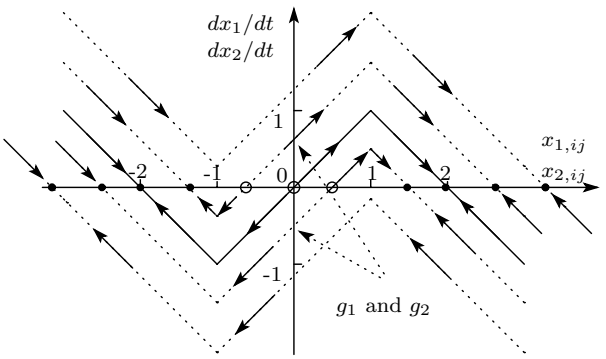


Fig. 4 The trajectories of state variables for various g_1 and g_2, where we set $a_{1,00} = a_{2,00} = 2$ and the stable and unstable equilibrium points are denoted by the solid dots and circles, respectively.

templates A_1, A_2, B_1, B_2, C_1 and C_2 are zero. For simplification, we can omit the subscript (i,j), because each cell in the CNNs is coupled in the same way. Thus, the cell equations are written as follows:

$$\left.\begin{aligned} \dot{x}_1 &= -x_1 + a_{1,00}y_1 + b_{1,00}u_1 + c_{1,00}y_2 + I_1 \\ &= -x_1 + a_{1,00}y_1 + g_1 \\ \dot{x}_2 &= -x_2 + a_{2,00}y_2 + b_{2,00}u_2 + c_{2,00}y_1 + I_2 \\ &= -x_2 + a_{2,00}y_2 + g_2 \end{aligned}\right\} \quad (4)$$

For the self-feedback coefficients $a_{1,00} > 1$ and $a_{2,00} > 1$, the trajectories of the state variables without g_1 and g_2 behave like as the solid line in Fig. 4. The trajectories of Eq. (4) can be obtained by shifting g_1 and g_2 values up and down as shown by the dotted lines in the figure. Observe that if we choose the initial conditions $x_1(0)$ and $x_2(0)$ larger than the unstable equilibrium points shown by circles, the steady-state output will be given by 1, and otherwise, the outputs will be given by -1. Thus, we have the following output relations depending on the initial conditions;

$$\left.\begin{aligned} y_1 &= \mathrm{sgn}\left[x_1(0) + \frac{g_1}{a_{1,00}-1}\right] \\ &= \mathrm{sgn}\left[x_1(0) + \frac{b_{1,00}u_1 + c_{1,00}y_2 + I_1}{a_{1,00}-1}\right] \\ y_2 &= \mathrm{sgn}\left[x_2(0) + \frac{g_2}{a_{2,00}-1}\right] \\ &= \mathrm{sgn}\left[x_2(0) + \frac{b_{2,00}u_2 + c_{2,00}y_1 + I_2}{a_{2,00}-1}\right] \end{aligned}\right\} \quad (5)$$

where $b_{1,00}$, $b_{2,00}$, $c_{1,00}$, $c_{2,00}$, I_1 and I_2 are the constant parameters. u_1 and u_2 are the inputs of cells, which are usually equal to 1 or -1 corresponding to black or white pixel in the binary images, or are between 1 and -1 for the gray scale images. Therefore, by suitably selecting these parameters and initial state conditions, we can obtain the expected outputs.

For example, consider the two-layer CNN to perform a linearly non-separable task—logic XOR. The logic function can be written as

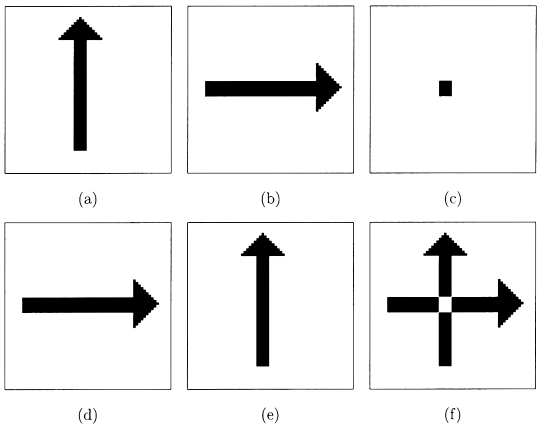


Fig. 5 An example for solving a linearly non-separable problem. (a), (b) and (c) are the input, the initial state condition and the output of the first layer, (d), (e) and (f) are the input, the initial state condition and the output of the second layer, respectively.

$$F = X_1 \oplus X_2. \quad (6)$$

In this logic, the input of two pixels has four possible combinations, and the output will be black only if one of the two pixels is black. The templates are:

$$A_1{=}A_2{=}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_1{=}B_2{=}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
$$C_2{=}\begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad C_1{=}0, \quad I_1{=}-1, \quad I_2{=}-1. \quad (7)$$

Consider an example for solving a linearly non-separable problem shown in Fig. 5. One of the two given images is set to the input u_1 of the first layer CNN and the initial state $x_2(0)$ of the second layer. Another image is set to the input u_2 of the second layer and the state $x_1(0)$ of the first layer. The execution result is obtained from the output of the second layer as shown by Fig. 5, where (a) and (d) are two input images of the two layers, and (b) and (e) are the initial states. The execution result is shown by (f), where the black pixels are obtained only when one of the two inputs has a black pixel, and white pixels are obtained for every other case. The output of the first layer shows the result obtained by Logic AND of the two images. These results can be calculated by the relations Eq. (5). From the above simulation result, we have shown that the simplified two-layer CNNs are capable to perform linearly non-separable tasks.

3.2 Compound Tasks

There are many complicated tasks such that we need to apply some steps of single-layer CNN operations such as Logic operation, edge detection and diffusion, etc.

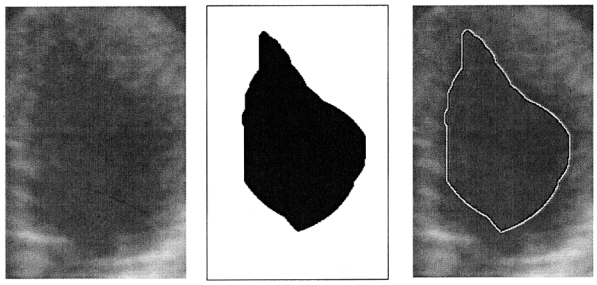


Fig. 6 Endocardial boundary detection (a) the initial states and inputs of both two layers, (b) the output of the first layer, (c)the output of the second layer.

in the step by step manner. Hence, these processes are tedious and complicated to choose their templates. We show in this section that it can be directly executed by the one step of a two-layer CNN.

3.2.1 Feature Detection of Endocardial Image

Medical image processing is one of major topics of CNNs [31]. The boundary detection, which is one of the feature extraction techniques, is of great importance both for qualitative and quantitative analysis for some medical image processing. Its image will help the doctors to correctly examine the patient's conditions. In this subsection, let us consider such an example that a two-layer CNN will detect the endocardial boundary of left ventricle from ultrasonic (US) image as shown in Fig. 6(a). Assume that we want to detect the endocardial boundary and superimpose it on the original image. This process can be more efficiently executed by the two-layer CNN in one step, compared to the single-layer CNNs [31].

Namely, the first layer CNN executes a threshold process with certainly diffusing and filtering. The second layer detects the edges from the first layer output with C_2 template and superimposes them on the original image. The final result can be obtained in the second layer output. The templates are given as follows:

$$A_1=\begin{bmatrix} 0 & 1.6 & 0 \\ 1.6 & 2 & 1.6 \\ 0 & 1.6 & 0 \end{bmatrix}, \quad B_1=0, \quad C_1=\begin{bmatrix} 0 & 0.5 & 0 \\ 0.5 & 2 & 0.5 \\ 0 & 0.5 & 0 \end{bmatrix},$$
$$A_2=\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_2=2, \quad C_2=\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix},$$
$$I_1=0, I_2=-1. \tag{8}$$

In this case, the original US image is set as both two layers as the initial states, and inputted to second layer as shown in Fig. 6(a). The input of the first layer may be arbitrary due to $B_1 = 0$. In the simulation, we set the first layer input as the same US image and applied the zero-fixed boundary condition to both two layers.

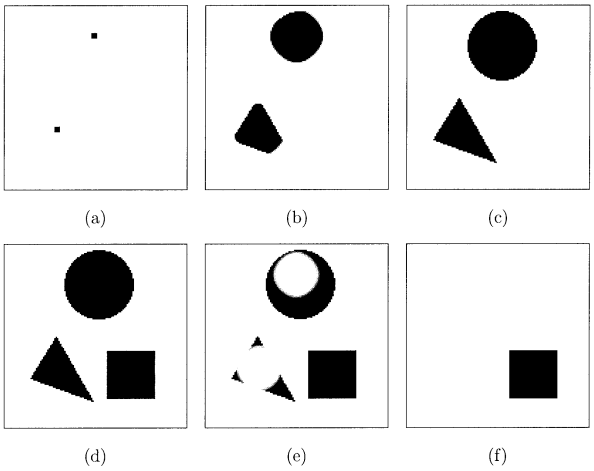


Fig. 7 An example for solving object separation problem. (a), (b) and (c) are the initial state condition, the transient result and the output of the first layer. (d) is the input of the first layer and the initial state condition of the second layer, (e) and (f) are the transient results and the output of the second layer, respectively.

Thus, we obtained the expected result Fig. 6(c) by superimposing the endocardial boundary on the original image. Figure 6(b) is binary endocardial image separated from the original image.

3.2.2 Object Separation

In this subsection, we consider another application to continuously extract a character from a word, and recognize it. After a character is extracted, it must be erased from the original text. This process can be recognized as an object separation. Now, let us consider a simple case separating the round and triangle objects from the three images of the round, rectangular and triangle as shown in Fig. 7(d). This task can be executed by two processes such as extracting the chosen objects by setting two pointers in advance as the initial states at the positions of the round and triangle objects in the first layer, and erasing the extracted objects from the original image at the same time in the second layer. This task can be similarly carried out by a single step of two-layer CNN as follows.

The original binary image shown by Fig. 7(d) is inputted to the first layer input and set as the initial state of the second layer. By setting two pointers in advance at the first layer as the initial state shown by Fig. 7(a), we can extract the specified objects under the consideration with recall template [15]. Its output is transferred to the second layer through C_2. Then, we can execute the logic difference at the second layer with logic difference template [15]. Thus, the extracted two objects are obtained by the first layer output, and the rest rectangle object is obtained by the second layer output as shown by Figs. 7(c) and (f), respectively. Figures 7(b) and (e) are the transient results of the first and second layers. In this example, we also have chosen zero-fixed

boundary condition [1]. The templates are

$$A_1 = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 2 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix},\ B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ C_1 = 0,$$
$$A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ B_2 = 0,\ C_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
$$I_1 = 0.25,\ I_2 = -1. \tag{9}$$

From the above two examples, we can conclude that the two-layer CNN can execute each compound task in one step, where each layer shares out the tasks and cooperates in each other. However, if we use single-layer CNNs, it will take two or more steps in the sequential manner. Thus, the two-layer CNN is more efficient compared to the single-layer CNNs.

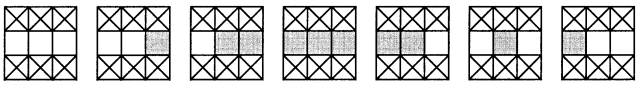
3.3 Complex Tasks

In many applications of image processing and pattern recognition, it is very important to detect the center point of a given object, because it is used as the reference position of the object. Unfortunately, the definition of the center point is very ambiguous, because there are various objects such as convex, concave, disk with white holes, etc. In this paper, we define "the center point is located at the halfway from the furthermost points of a given object image." Therefore, at first, a given object image is changed into a rectangular block including a given object with shadow template [15]. Thus, the center point problem is reduced to the problem of finding out the middle point of the rectangular object. If a single-layer CNN is applied to the center point problem, we need to progressively peel off one column or row pixels from four directions—left, bottom, right and up hand sides. These sequential processes are continued to get the center point. The number of iterations depends on the size of the object. This is a very complicated process. If we use a two-layer CNN, this task can be solved by two steps. The first step is to detect the center line of the object, and the second step is to detect the center point of the above center line.

3.3.1 Detection of the Center Line of Object

Now, consider a center line detection problem shown in Fig. 9(a). The object image is set to the initial states of both CNN layers as shown by Figs. 9(a) and (g). We used the following templates to simultaneously peel off the most left and right hand side pixels:

$$A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 2 & 0.5 \\ 0 & 0 & 0 \end{bmatrix},\ C_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2.8 & -2 \\ 0 & 0 & 0 \end{bmatrix},\ \begin{matrix} B_1 = 0, \\ I_1 = -2, \end{matrix}$$



(a) (b) (c) (d) (e) (f) (g)

Fig. 8 All possible landscapes of a cell.

Table 1 The landscape and stable equilibrium point of a cell in the first layer.

Landscape	$g_{1,ij}$	initial state	stable output
case (a)	−3.8	−1	−1
case (b)	−6.8	−1	−1
case (c)	−1.2	1	−1
case (d)	−0.2	1	1
case (e)	2.8	1	1
case (f)	1.8	1	1
case (g)	−2.8	−1	−1

$$A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 2 & 0.5 \\ 0 & 0 & 0 \end{bmatrix},\ C_2 = \begin{bmatrix} 0 & 0 & 0 \\ -2 & 2.8 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ \begin{matrix} B_2 = 0, \\ I_2 = -2, \end{matrix} \tag{10}$$

The cell's state equations can be rewritten in the following forms:

$$\left.\begin{aligned} \dot{x}_{1;ij} &= -f_1(x_{1;i,j}) + g_1 \\ \dot{x}_{2;ij} &= -f_2(x_{2;i,j}) + g_2 \end{aligned}\right\} \tag{11}$$

where

$$\begin{aligned} f_1(x_{1;i,j}) &= x_{1;i,j} - (|x_{1;i,j}+1| - |x_{1;i,j}-1|), \\ f_2(x_{2;i,j}) &= x_{2;i,j} - (|x_{2;i,j}+1| - |x_{2;i,j}-1|), \\ g_{1;i,j} &= 0.5y_{1;i,j-1} + 0.5y_{1;i,j+1} + 2.8y_{2;i,j} - 2y_{2;i,j+1} - 2, \\ g_{2;i,j} &= 0.5y_{2;i,j+1} + 0.5y_{2;i,j-1} - 2y_{1;i,j-1} + 2.8y_{1;i,j} - 2. \end{aligned} \tag{12}$$

To analyze the stability of the equilibrium point of a cell, we restrict the discussion to a landscape of the cell by listing all those neighboring states, including itself. Since the two state $x_1(0)$ and $x_2(0)$ are the same at the beginning of this simulation, the landscape can be divided into seven possible cases as shown in Fig. 8, in which the black squares denote cells having value 1 and the white squares −1 valued cells. On the other hand, the crossed square cells stand for 'don't care' for the center cell, because the template coefficients in Eq. (10) corresponding to those positions are zeros.

For the case of Fig. 8(a), we have

$$\begin{aligned} g_{1;i,j} &= 0.5y_{1;i,j-1} + 0.5y_{1;i,j+1} + 2.8y_{2;i,j} - 2y_{2;i,j+1} - 2 \\ &= 0.5\times(-1) + 0.5\times(-1) + 2.8\times(-1) - 2\times(-1) - 2 \\ &= -3.8. \end{aligned}$$

In this case, we have $y_1 = \mathrm{sgn}[x_1(0) - 3.8]$ from the relations (5). Thus, we have $y_1 = -1$ (white) for $x_1(0) \leq -1$. Similarly, for the other cases from Figs. 8(b) to (g), their stable equilibrium points can be obtained as listed in Table 1. Observe from the table that the output of the center cell only in the case (c) will be changed from

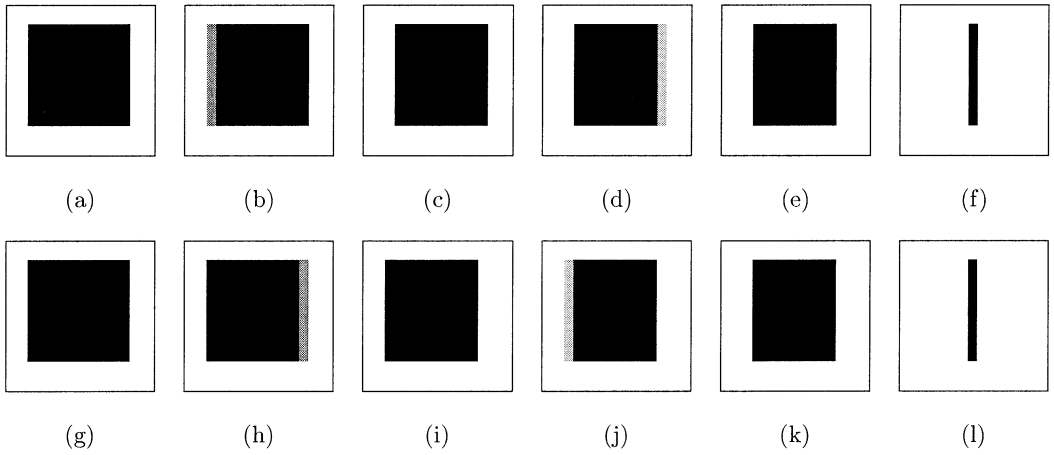


Fig. 9 An example for extracting the center line of an object. States from (a) to (f) are the output of the first layer in the time progress, and those from (g) to (l) are the output of second layer. Final results of the two layer CNN are shown by (f) and (l).

1 (black) to −1 (white), and unchanged for the other cases. Thus, the first layer of the CNN performs a peeling-off of the leftmost pixels of the object as shown in Fig. 9(b). By the similar analysis, the second layer performs a peeling-off of the rightmost pixels of the object as shown in Fig. 9(h). These peelings are executed at the same time in each layer. Thus, we get the middle results as shown by Figs. 9(c) and (i), respectively. After that, the landscape can be divided into eight possible cases as shown in Fig. 10. At this moment, we have $g_{1;i,j} = -2.8$ and $y_1 = \mathrm{sgn}[x_1(0) - 2.8] = -1$ for case Fig. 10(g), $g_{2;i,j} = -2.8$ and $y_2 = \mathrm{sgn}[x_2(0) - 2.8] = -1$ for case Fig. 10(c). Thus, both the rightmost pixels of the first layer and the leftmost pixels of the second layer are peeled off as shown in Figs. 9(d) and (j), respectively. Figures 9(e) and (k) are the same as (a) and (g), except for the both end pixels which are peeled off. Thus, one cycle of the peeling-off is completed. This cycle continues until the vertical centerline is remained, where the transient has arrived at the steady state corresponding to Fig. 9(f). This dynamic process can be observed from the graph of the sum of State-error Square varying with the integration time shown in Fig. 11. The sum of State-error Square is defined as

$$S_{CNN}(t) = \sum_{i=1}^{M}\sum_{j=1}^{N}((x_{1,ij}(t+\Delta t) - x_{1,ij}(t))^2 + (x_{2,ij}(t+\Delta t) - x_{2,ij}(t))^2) \qquad (13)$$

The peaks show the states when the pixels from the two layers are unbalanced states. When the object is completely peeled into center line, the sum of state-errors square decays to zero. The two layer stable outputs are shown by Figs. 9(f) and (l), respectively. In this simulation, the zero-fixed boundary condition is also adopted.

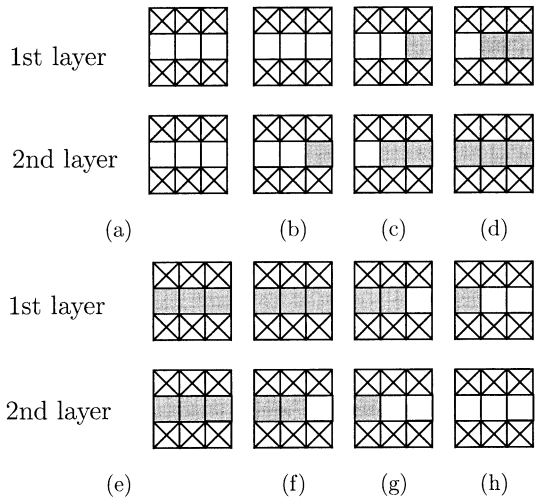


Fig. 10 All possible landscapes of a cell at a transient.

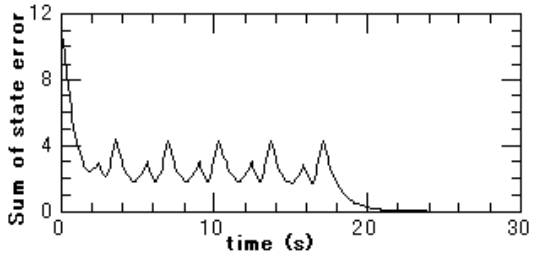


Fig. 11 The sum of State-error Square via time for center line extraction.

3.3.2 Detection of the Center Point from the Above Line

To detect the center point from the above center line, we need to apply the same algorithm to the center line with the transposed templates as follows:

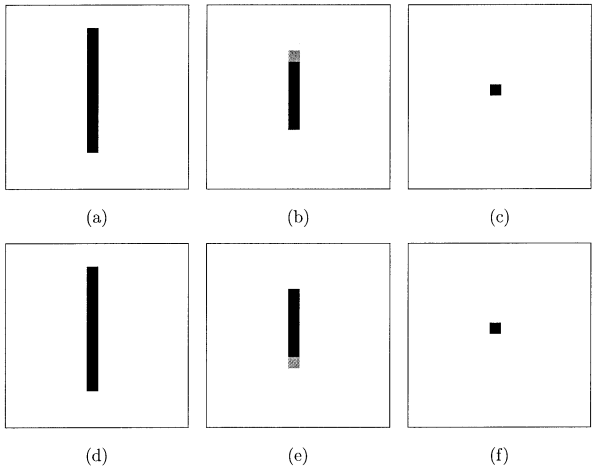


Fig. 12 An example for extracting the center point of line. (a), (b) and (c) are the initial state condition, the transient result and the output of first layer. (d), (e) and (f) are the initial state condition, the transient result and the output of the second layer, respectively.

$$A_1=\begin{bmatrix}0 & 0.5 & 0\\ 0 & 2 & 0\\ 0 & 0.5 & 0\end{bmatrix},\ C_1=\begin{bmatrix}0 & 0 & 0\\ 0 & 2.8 & 0\\ 0 & -2 & 0\end{bmatrix},\ \begin{matrix}B_1=0,\\ I_1=-2,\end{matrix}$$
$$A_2=\begin{bmatrix}0 & 0.5 & 0\\ 0 & 2 & 0\\ 0 & 0.5 & 0\end{bmatrix},\ C_2=\begin{bmatrix}0 & -2 & 0\\ 0 & 2.8 & 0\\ 0 & 0 & 0\end{bmatrix},\ \begin{matrix}B_2=0,\\ I_2=-2,\end{matrix} \quad (14)$$

The transient behaviors are shown in Fig. 12. Thus, we can find the center point in two steps.

4. Stability of Two-Layer Cellular Neural Networks

Since the CNNs are usually required to be stable for the image processing applications, the studies on complete stability have been vigorously discussed and many criteria have been obtained [1], [23]–[26]. In this section, we will discuss the convergence property, and its related problems for the two-layer CNNs with symmetric and/or special coupling templates.

For analyzing the convergence properties of dynamic nonlinear systems, one of the most effective techniques is Lyapunov's method, which has been successfully applied to the stability analysis of single-layer CNN [1]. In this section, we analyze the stability of the two-layer CNN with the same techniques. We define a Lyapunov function $E(t)$ of a two-layer CNN by the scalar function, which is similar to the one used in [1], [22] and can be interpreted as the "generalized energy" for a two-layer CNN.

$$\begin{aligned}E(t) =& -\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)} A_1(i,j;k,l)y_{1,ij}y_{1,kl}\\ &+\sum_{(i,j)}\sum_{(k,l)} B_1(i,j;k,l)y_{1,ij}u_{1,kl}\\ &-\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)} C_1(i,j;k,l)y_{1,ij}y_{2,kl}\\ &+\frac{1}{2}\sum_{(i,j)} y_{1,ij}^2-\sum_{(i,j)} I_1y_{1,ij}\\ &-\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)} A_2(i,j;k,l)y_{2,ij}y_{2,kl}\\ &+\sum_{(i,j)}\sum_{(k,l)} B_2(i,j;k,l)y_{2,ij}u_{2,kl}\\ &-\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)} C_2(i,j;k,l)y_{2,ij}y_{1,kl}\\ &+\frac{1}{2}\sum_{(i,j)} y_{2,ij}^2-\sum_{(i,j)} I_2y_{1,ij} \end{aligned} \quad (15)$$

Therefore, we can show that the function $E(t)$ is bounded as follows:

$$\max_t |E(t)| \le E_{max} \quad (16)$$

Where

$$\begin{aligned}E_{max} =& +\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)}|A_1(i,j;k,l)|\\ &+\sum_{(i,j)}\sum_{(k,l)}|B_1(i,j;k,l)|\\ &+\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)}|C_1(i,j;k,l)|\\ &+\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)}|A_2(i,j;k,l)|\\ &+\sum_{(i,j)}\sum_{(k,l)}|B_2(i,j;k,l)|\\ &+\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)}|C_2(i,j;k,l)|\\ &+MN(1+|I_1|+|I_2|)\end{aligned} \quad (17)$$

Now, let us assume the templates satisfy the following condition Eq. (18):

$$\left.\begin{aligned}A_1(i,j;k,l)&=A_1(k,l;i,j)\\ A_2(i,j;k,l)&=A_2(k,l;i,j)\\ C_1(i,j;k,l)&=C_2(k,l;i,j)\\ C_2(i,j;k,l)&=C_1(k,l;i,j)\end{aligned}\right\} \quad (18)$$

From Eq. (15), we have the following relation:

$$\frac{dE(t)}{dt}=-\sum_{(i,j)}\left[\left(\frac{dy_{1,ij}}{dt}\right)^2+\left(\frac{dy_{2,ij}}{dt}\right)^2\right]\le 0 \quad (19)$$

Thus, we found that the energy function is monotone decreasing, where we used the following constraint conditions:

$$\frac{dy_{1,ij}}{dx_{1,ij}}=\begin{cases}1 & |x_{1,ij}|<1\\ 0 & |x_{1,ij}|\ge 1\end{cases} \quad (20)$$

$$x_{1,ij} = y_{1,ij},\ |x_{1,ij}| < 1 \tag{21}$$

$$\frac{dy_{2,ij}}{dx_{2,ij}} = \begin{cases} 1 & |x_{2,ij}| < 1 \\ 0 & |x_{2,ij}| \geq 1 \end{cases} \tag{22}$$

$$x_{2,ij} = y_{2,ij},\ |x_{2,ij}| < 1 \tag{23}$$

We found that from Eqs. (16) and (19), for any given inputs u_1, u_2 and the initial states x_1, x_2, we obtain that

$$\lim_{t\to\infty} E(t) = const. \tag{24}$$

and

$$\lim_{t\to\infty} \frac{dE(t)}{dt} = 0 \tag{25}$$

Thus, under the condition Eq. (18), we always have the stable steady state outputs in both two layers after the transient. Now, we show that both the states x_1 and x_2 approaches to the equilibrium points. Let us consider the system equations Eqs. (1)–(3) again, and rewrite them in the following form:

$$\left.\begin{aligned} \frac{dx_{1,ij}(t)}{dt} &= -f_1(x_{1,ij}(t)) + g_1(t) \\ \frac{dx_{2,ij}(t)}{dt} &= -f_2(x_{2,ij}(t)) + g_2(t) \end{aligned}\right\} \tag{26}$$

where

$$f_1(x_{1,ij}) = x_{1,ij} - \frac{1}{2}A_1(i,j;i,j)(|x_{1,ij}+1| - |x_{1,ij}-1|)$$

$$f_2(x_{2,ij}) = x_{2,ij} - \frac{1}{2}A_2(i,j;i,j)(|x_{2,ij}+1| - |x_{2,ij}-1|)$$

$$g_1(t) = \sum_{(k,l)\neq(i,j)} A_1(i,j;k,l)y_{1,kl} + \sum_{(k,l)} C_1(i,j;k,l)y_{2,kl} + \sum_{(k,l)} B_1(i,j;k,l)u_{1,kl} + I_1$$

$$g_2(t) = \sum_{(k,l)\neq(i,j)} A_2(i,j;k,l)y_{2,kl} + \sum_{(k,l)} C_2(i,j;k,l)y_{1,kl} + \sum_{(k,l)} B_2(i,j;k,l)u_{2,kl} + I_2$$

Observe that $g_1(t)$ and $g_2(t)$ are not related to the states $x_{1,ij}$ and $x_{2,ij}$, and are only the functions of the outputs (y_1, y_2), inputs (u_1, u_2) and biases (I_1, I_2). Therefore, the trajectories of Eq. (26) for $A_1(i,j;i,j) > 1$ and $A_2(i,j;i,j) > 1$ have the same structures as shown in Fig. 4, and have the stable and unstable equilibrium points. Thus, we can conclude for the condition Eq. (18) that the steady state of two-layer CNN is completely stable, in the meaning that both states and outputs are in the steady state. The validity of the above analysis has been verified by the above simulations shown in Sects. 3.3.1 and 3.3.2.

5. Conclusions

In this paper, we have proposed a two-layer cellular neural network model, and discussed the applications and stability. We found that it has many interesting applications such as linearly non-separable task—logic XOR, some compound tasks and complex problems—the center point detection of the object. Although, of course, they can be solved by single-layer CNNs, the applications of the two-layer CNNs to these problems are very efficient compared to the single-layer CNNs. Especially, the center point detection problem with the two-layer CNNs can be found the solution in two steps, however, if we use the single-layer CNNs, it will be found many steps depending on the object size in serial manner. We can prove the stability for the special cases of two-layer CNNs, where its necessary conditions are proved by the Lyapunov function and behaviors of the trajectories on the phase plane. In future problems, we want to find the special applications of the two-layer CNNs such that they cannot be solved by the single-layer CNNs.

References

[1] L.O. Chua and L. Yang, "Cellular neural networks: Theory," IEEE Trans. Circuits & Syst., vol.35, no.10, pp.1257–1272, Oct. 1988.
[2] L.O. Chua and L. Yang, "Cellular neural networks: Applications," IEEE Trans. Circuits & Syst., vol.35, no.10, pp.1273–1290, Oct. 1988.
[3] M. Ikegami and M. Tanaka, "Image coding and decoding by discrete time cellular neural networks and its error evaluation," IEICE Trans. Fundamentals (Japanese Edition), vol.J77-A, no.7, pp.954–964, July 1994.
[4] M. Tanaka, K.R. Crounse, and T. Roska, "Parallel analog image coding and decoding by using cellular neural networks," IEICE Trans. Fundamentals, vol.E77-A, no.7, pp.1387–1395, July 1994.
[5] M. Ohnishi, M. Ikegami, Y. Tanji, K. Jinno, and M. Tanaka, "Image intensity conversion using discrete time cellular neural networks," J. Signal Processing, vol.3, no.6, pp.491–498, Nov. 1999.
[6] C. Rekeczky, A. Ushida, and T. Roska, "Rotation invariant detection of moving and standing objects using analogic cellular neural network algorithms based on ring-codes," IEICE Trans. Fundamentals, vol.E78-A, no.10, pp.1316–1330, Oct. 1995.
[7] T. Roska and L.O. Chua, "The CNN universal machine," IEEE Trans. Circuits & Syst., vol.40, no.3, pp.163–173, March 1993.
[8] L.O. Chua and T. Roska, "The CNN pradigm," IEEE Trans. Circuits & Syst., vol.40, no.3, pp.147–156, March 1993.
[9] L.O. Chua and T. Roska, "The CNN universal machine: An analogic array computer," IEEE Trans. Circuits & System II, vol.40, no.3, pp.163–173, March 1993.
[10] F. Werblin, T. Roska, and L.O. Chua, "The analogic cellular neural network as bionic eye," Int. J. Circuit Theory & Appl., vol.23, pp.541–569, 1995.
[11] P.L. Ventianer, F. Werblin, T. Roska, and L.O. Chua, "Analogic CNN algorithms for some image compression

and restoratin tasks," IEEE Trans. Circuits & Syst., vol.42, no.5, pp.278–284, May 1995.
[12] H. Harrer and J.A. Nossek, "Skeletonization: A new application for discrete-time cellular neural networks using time-variant templates," Proc. IEEE Int. Symp. Circuits & Syst., pp.2897–2900, 1992.
[13] D. Yu, C. Ho, X. Yu, and S. Mori, "On the application of cellular automata to image thinning with cellular neural network," Proc. IEEE Int. Wkshp. Cellular Neural Networks and Their Applicat, CNNA-92, pp.210–215, 1992.
[14] T. Matsumoto, T. Yokohama, H. Suzuki, R. Furukawa, A. Oshimoto, T. Shimmi, Y. Matsushida, T. Seo, and L.O. Chua, "Several image processing examples by CNN," Proc. IEEE Int. Wkshp. Cellular Neural Networks and Their Applicat., CNNA-90, pp.100–112, 1990.
[15] T. Roska, L. Kek, L. Nemes, A. Zarandy, and P. Szolgay, "CNN software library (Templates and algorithms)," Analogical and neural computing laboratory, Computer and Automation Institute of the Hungarian Academy of Sciences, Version 7.3, 1999.
[16] L.O. Chua, M. Hasler, G.S. Mochytz, and J. Neirynck, "Autonomous cellular neural networks: A unified paradigm for pattern formation and active wave propagation," IEEE Trans. Circuits & Syst., vol.42, no.10, pp.559–577, 1995.
[17] Z.H. Yang, Y. Nishio, and A. Ushida, "A model of nonlinear phenomena: A unified framework two layer CNN," Proc. the 2001 international technical conference on circuits/systems, computer and communications, pp.1288–1291, 2001.
[18] L. Goras, L.O. Chua, D.M.W. Leenaerts, and L. Pivka, "Turing patterns in CNNs part I–III," IEEE Trans. Circuits & Syst., vol.42, no.10, pp.602–637, Oct. 1995.
[19] M.J. Ogorzalek, Z. Galias, A.M. Dabrowski, and W.R. Dabrowki, "Chaotic waves and spatio-temporal patterns in large arrays of doubly-coupled Chua's circuits," IEEE Trans. Circuits & Syst., vol.42, no.10, pp.706–714, 1995.
[20] L. Pivka, "Autowave and spatio-temporal chaos in CNNs Part I and II: A tutorial," IEEE Trans. Circuits & Syst., vol.42, no.10, pp.638–664, 1995.
[21] P. Thiran, K.R. Crounse, L.O. Chua, and M. Hasler, "Pattern formation properties of autonomous cellular neural networks," IEEE Trans. Circuits & Syst., vol.42, no.10, pp.757–774, 1995.
[22] J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," Proc. Natl. Acad. Sci. USA, vol.81, pp.3088–3092, 1984.
[23] C.W. Wu and L.O. Chua, "More rigorous proof of complete stability of cellular neural networks," IEEE Trans. Circuits & Syst. I, vol.44, no.4, pp.370–371, 1997.
[24] L.O. Chua and T. Roska, "Stability of a class of nonreciprocal cellular neural networks," IEEE Trans. Circuits & Syst., vol.37, no.12, pp.1520–1527, 1990.
[25] L.O. Chua and C.W. Wu, "On the universe of stable cellular neural networks," Int. J. Circuit Theory & Appl., vol.20, pp.497–517, 1992.
[26] N. Takahashi and L.O. Chua, "On the complete stability of nonsymmetric cellular neural networks," IEEE Trans. Circuits & Syst. I, vol.45, no.7, pp.754–758, 1998.
[27] P. Arena, S. Baglio, L. Fortune, and G. Manganaro, "Complexity in a two-layer CNN," Proc. IEEE Int. Wkshp. Cellular Neural Networks and Their Applicat., CNNA-96, pp.127–132, 1996.
[28] C.W. Wu, L.O. Chua, and T. Roska, "A two-layer radon transform cellular neural network," IEEE Trans. Circuits & Syst. II: Analog and Digital Signal Processing, vol.39, no.7, pp.488–489, 1992.
[29] C. Botoca, "An associative memory using a two layer cellular neural network, Part I," Proc. International Conference SCS 95, pp.121–124, Oct. 1995.
[30] B. Siemiatkowska, "A highly parallel method for mapping and navigation of an autonomous mobile robot," Proc. International Conference on Robotics and Automation, pp.2796–2801, San Diego, California, May 1994.
[31] C. Rekeczky, A. Tahy, Z. Vegh, and T. Roska, "CNN-based spatio-temporal nonlinear filtering and endocardial boundary detection in echocardiography," Int. J. Circuit Theory & Appl., vol.27, pp.171–207, 1999.
[32] J.M. Zurada, Introduction to Artificial Neural Systems, West Publishing Company, St. Paul, 1992

Zonghuang Yang received the B.E. degree in Radio Engineering from South East University, Nanjing China in 1985, and the M.E. degree in Electrical and Electronic Engineering from Tokushima University, Tokushima Japan, in 2001. He is currently working toward the Ph.D. degree at same department of Tokushima University. His research interests include theory and application of cellular neural networks.

Yoshifumi Nishio received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama Japan, in 1988, 1990, and 1993, respectively. In 1993, he joined the Department of Electrical and Electronic Engineering at Tokushima University, Tokushima Japan, where he is currently an Associate Professor. From May 2000 he spent a year in the Laboratory of Nonlinear Systems (LANOS) at the Swiss Federal Institute of Technology Lausanne (EPFL) as a visiting professor. His research interests include analysis and application of chaos in electrical circuits, analysis of synchronization in coupled oscillators circuits, development of analyzing methods for nonlinear circuits and theory and application of cellular neural networks. Dr. Nishio is a member of the IEEE.

Akio Ushida received the B.E. and M.E. degrees in Electrical Engineering from Tokushima University in 1961 and 1966, respectively, and the Ph.D. degree in Electrical Engineering from University of Osaka Prefecture in 1974. He was an associate professor from 1973 to 1980 at Tokushima University. since 1980, he has been a professor in the Department of Electrical and Electronic Engineering at Tokushima University. From 1974 to 1975 he spent one year as a visiting scholar at the Department of Electrical Engineering and Computer sciences at the University of California, Berkeley. His current research interests include numerical methods and computer-aided analysis of nonlinear systems. Dr. Ushida is a member of the IEEE.