Errorless reproduction of given pattern dynamics by means of cellular automata

Teruhiko Tamura,[1] Jousuke Kuroiwa,[2] and Shigetoshi Nara[3]
[1]*The Graduate School of Natural Science & Technology, Okayama University, 700-8530 Okayama, Japan*
[2]*Faculty of Integrated Arts & Sciences, Hiroshima University, 739-8521 Higashi-Hiroshima, Japan*
[3]*Department of Electrical & Electronic Engineering, Faculty of Engineering, Okayama University, 700-8530 Okayama, Japan*
(Received 18 October 2002; revised manuscript received 16 May 2003; published 29 September 2003)

In this paper we propose the two methods to reproduce given binary pattern dynamics with cellular automata. The point is that one can easily find a sequence of rules or specified rules in two-state multineighbors cellular automata, which enable an errorless description and reproduction of given multiple sequences of binary patterns. Actual examples using computer experiments for one-dimensional bit-pattern data (digital sound signals, multiple sequences of cycle patterns) are given. Noise robustness and the other important dynamical properties of these methods are investigated from the perspective of "rule dynamics" and in comparison with a recurrent neural network model, which enables us to embed given binary patterns as multiple attractors in the form of fixed points or limit cycles.

DOI: 10.1103/PhysRevE.68.036707 PACS number(s): 95.75.Pq, 05.45.−a, 43.38.+n, 89.75.Kd

I. INTRODUCTION

Biological systems have been attracting a lot of interest because of their excellent functions that work under various environments. Traditionally, the main methodologies for scientifically approaching their mechanisms have been physics, chemistry, and so on. In contrast, however, the last few decades have seen that new approaches appear associated with the remarkable development of computers and simulation methods. Furthermore, the discovery of chaotic dynamics in biological systems, including the brain, has produced a big impact and also has been attracting great interest from many scientists with the question of "what is the role of chaos in biological systems?" However, the great complexity originating from dynamics in systems with large but finite degrees of freedom such as biological systems is still preventing us from understanding the scientific mechanisms of their excellent information or control functions, in spite of much effort by ambitious researchers in a variety of scientific fields [1–13].

In these situations, we believe that chaotic dynamics, even in systems with many degrees of freedom, could be generated by a single or a sequence of certain simple deterministic rules, as observed in systems with few degrees of freedom [7,8]. By virtue of this, one could gain a considerable understanding of the dynamical mechanisms of their functioning, which could be applied to realizing complex controls or complex information processing via a certain simple rule [10–13].

Based on these motivations, we focused on describing and reproducing high-dimensional chaotic dynamics by utilizing a certain set of rules or a sequence of simple rules, which could be regarded as an inverse problem in nonlinear dynamics. Cellular automata (CA) is one such method and has been intensively studied, for instance, in [7–9]. Recently, we focused on real digital sound data (spoken-word data and music data) as an example of chaotic dynamics, and have proposed a method for describing them by rule dynamics of one-dimensional CA with the two states and three neighbors [14–16]. In particular, Ref. [16] had shown that, with the use of only two rules, perfectly reproducible coding of digital sound data by CA rule dynamics is possible for a great deal of spoken-word data and music data formatted and used in a standard compact disk. It should be noted that our coding method is associated with data compression without any loss of information.

In this paper, extending our viewpoint more generically, we report the dynamical properties produced by this method in more detail, including a discovery that each rule sequence works as a generator of attractor dynamics for arbitrarily given initial patterns, where the word "attractor" is used in a little nonconventional meaning which will be mentioned in a later section (Sec. II A). Furthermore, we would like to propose another method for reproducing multiple sequences of cycle patterns that are rather long data strings represented by bit patterns. We shall briefly investigate dynamical properties and noise robustness, particularly in comparison with a recurrent neural network model.

II. DYNAMICAL PROPERTIES OF AN ERRORLESS DESCRIPTION OF DIGITAL SOUND DATA BY ONLY TWO CA RULES

A. Errorless reproduction of sound data

Let us briefly introduce CA and our method that considers one-dimensional two-state–three-neighbor cellular automata (abbreviated as 1-2-3 CA hereafter), where each cell is arranged on a one-dimensional chain. We employ variables a_i^t ($=0$ or 1, $i=1,\ldots,N$) which indicate the state of the ith site in chain at time step t. The state of the ith site at time step $t+1$, a_i^{t+1} is determined by the states of itself and those of the neighboring two sites at time step t, so that the updating rule can be represented as

$$a_i^{t+1}=f(a_{i-1}^t,a_i^t,a_{i+1}^t), \quad (1)$$

where the function $f(\cdot)$ is called a transition function, which updates the state of a_i^t to a_i^{t+1}. To specify updating functions, we introduce the following abbreviations:

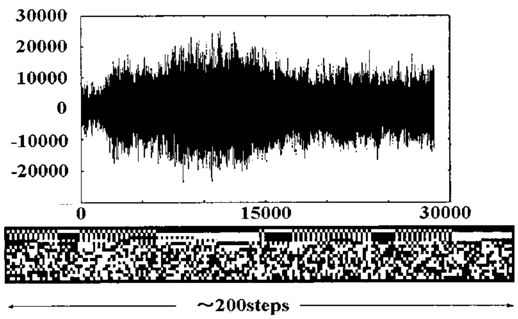

FIG. 1. An example of sound data taken from music (upper) and 16 bit-pattern sequences (lower) which correspond to digital data sampled and quantized according to the standard musical data format of CDs (compact discs), i.e., a frequency of 44.1 kHz and 16 bits, so that the horizontal axis is in the unit of time step 1/44100 s and the vertical axis is in the dimensionless scale of 2^{16} with the maximum amplitude unit.

$$\begin{aligned} f(0,0,0)&=f_0, \quad f(0,0,1)=f_1,\\ f(0,1,0)&=f_2, \quad f(0,1,1)=f_3,\\ f(1,0,0)&=f_4, \quad f(1,0,1)=f_5, \qquad (2)\\ f(1,1,0)&=f_6, \quad f(1,1,1)=f_7, \end{aligned}$$

where $f_i=0$ or 1 $(i=0,\ldots,7)$. By choosing each f_i to be 0 or 1, we can determine a certain specified rule. If we assign the two states of each cell to 0 or 1, respectively, the time development of a certain initial state in the 1-2-3 CA gives a sequence of one-dimensional bit patterns consisting of 0 or 1 [19]. By introducing a number as $n=2^0f_0+2^1f_1+2^2f_2+2^3f_3+2^4f_4+2^5f_5+2^6f_6+2^7f_7$, one can name each n "the specified rule number" from 0 to 255 (totally $2^8=256$ rules in 1-2-3 CA). Now, let us briefly present our basic idea for describing digital sound data by rule dynamics of the 1-2-3 CA [14]. In modern technologies, analog signals are often recorded as digital signals both in time and amplitude following sampling and analog-digital transformation. For instance, in the standard musical data format of CDs (compact discs), signals are sampled at a frequency of 44.1 kHz and 16-bits amplitude quantization. An example is shown in Fig. 1. Digital sound data are also represented as a sequence of binary pattern strings consisting of 16 bits, which can be regarded as the time development of a corresponding 1-2-3 CA consisting of 16 cells.

The problem is how to determine a rule sequence of the 1-2-3 CA that can reproduce the sequence of 16 bit patterns that describes the digital sound signals.

In the previous papers [14,15], all 256 rules were applied once to each successive step and the best rule to reproduce original binary data pattern was chosen. In most of the steps, however, original data cannot be reproduced without a certain error tolerance. Thus, the second proposed method is "repeated applications of rules" and we successfully found that errorless reproduction of original binary patterns is possible with only two rules of 1-2-3 CA, for instance, (51, 240) or (90, 180), and so on. We had shown that there

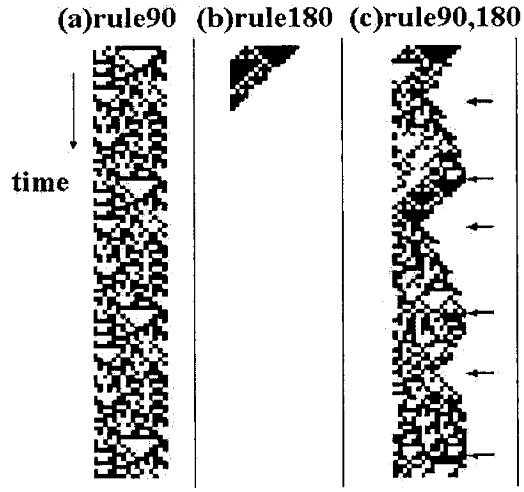

FIG. 2. Left: pattern dynamics of 16 cells starting from a certain initial condition where the rule number is 90. Middle: the same initial condition but the rule number is 180. Right: an actual example of sound data description by 1-2-3 CA, where only the two rules, 90 and 180, are used. The arrows indicate the 16-bit patterns corresponding to the sound data, where the intermediate patterns are these produced by the two-rule sequence, which perfectly reproduces the original sound data by sequentially applying the two rules. It should be noted that gray cells and black cells correspond to the rule numbers, 90 (gray) and 180 (black), respectively.

are many cases that exhibit not only perfect reproduction, but also result in data compression (see [16]). An example of a sequence of patterns that describes a sound signal without error is shown in Fig. 2(c) (only ~50 steps for saving space of showing).

Furthermore, we have discovered that the rule sequence does not care for the existence of noise. For instance, an initial condition including one-bit noise surely converges into the original pattern sequence by applying the rule sequence (see Fig. 3). Even starting from arbitrary given initial patterns, the sequence of original patterns can be recovered by applying the sequence of two rules. This means that the sequence of rules for reproducing given pattern dynamics works as a generator of "attractor dynamics." One would guess that, in a conventional interpretation, "attractor" is meaningful only when phenomena occur in autonomous systems—i.e., systems in which the evolution rules do not change as time goes on. In the present case, although the updating of states is associated with changing rules, there is the remarkable dynamical property that dynamics removes noise and results in converging into a specified pattern dynamics. It could be interpreted as "attractor dynamics" in our work and we use it in later descriptions.

B. Attractor dynamics generated by sequences of two rules to give perfect reproduction of sound data

As noted in the previous section, one-bit difference (noise) expands slightly during the initial stage of updating, resulting in the pattern dynamics becoming slightly different from the original pattern sequence. However, in rather short time steps, it returns to the original pattern sequence. We

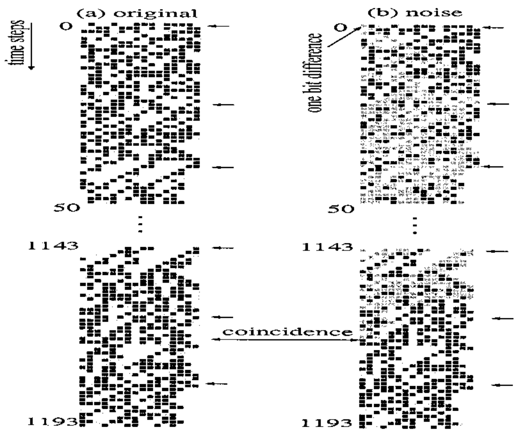

FIG. 3. (a) The original pattern sequence that describes the sound signal perfectly, where the data are taken from 1–50 steps. The gray stripe corresponds to the sound amplitude, while the other patterns are intermediate ones generated by multiple application of the two rules, 90 and 180. (b) The original pattern sequence from 1143 steps to 1193 steps. (c) The pattern sequence starting from an initial pattern having noise, in the case of (a). Note that one-bit difference extends and the pattern dynamics becomes totally different to (a). (d) The same pattern as (c), where the cells having different states to the original pattern sequence are shown in blue color (neglect this figure if the paper is published in noncolor printing). (e) The pattern sequence starting from the same initial condition as (c), where the steps from 1143–1193 are shown. It is important to note that the differences vanish, which means that the noise vanishes and the updated patterns return to the original bit patterns.

have confirmed by the computer experiment the fact that even when all of 2^{16} patterns are taken as initial conditions, they all converge to the pattern sequence of the corresponding sound data, so that the rule sequence only works as one strong attractor in the state space. This is one of the crucially important points of the present paper.

Now, to study a little more in detail how the converging dynamics recover the original pattern sequence, let us show, in Fig. 4, the distributions of step numbers until the updating of bit patterns returns the original one. Figure 5 shows the

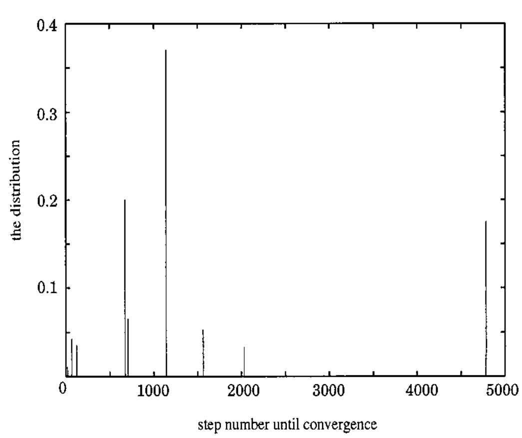

FIG. 4. Distribution of the step numbers until the updating of patterns converges with the original sound data patterns, where the rule sequence of two rules, in this case, 90 and 180, are applied.

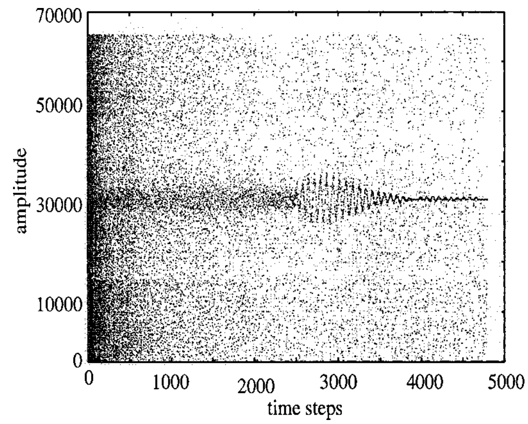

FIG. 5. Distribution of amplitude until the updating of patterns converges with the original sound data patterns, where the rule sequence of two rules, in this case, 90 and 180, are applied.

distributions of amplitude values that are generated during the update of bit patterns until they return the corresponding sound data patterns, where in the latter, bit patterns are converted to actual integer values represented by 16-bits at each time step.

From these two figures and our simulation results, we can gain an insight into the process of attracting dynamics generated by the rule sequences corresponding to each sound signal.

(1) In most cases, the number of steps taken until the state updating converges to the recorded signals is, at most, from several hundred to a few thousand. Thus the convergence is quite rapid in the sense that the length of each pattern sequence obtained from only a 1 s sound signal consists of 44 100 steps in the present experiment. This means that the attracting force generated by the rule sequence is quite strong.

(2) From the rough evaluation of Fig. 5, one can observe that there are not many paths to the attractors because the density of points quickly decreases as the patterns are updated. This means that the 2^{16} initial patterns quickly converge during updating within a rather small number of time steps.

(3) The statement of (2) is also confirmed by the fact that the distribution of step numbers until the updated patterns return the corresponding attractor (the sound signal) is considerably localized to a small number of steps. This means that updated patterns quickly converge into the same sequence of patterns.

III. ERRORLESS REPRODUCTION OF MULTIPLE CYCLIC BINARY PATTERN SEQUENCES BY TWO-STATE MANY-NEIGHBORS CA

A. Totalistic rule and errorless description in symmetric neighboring cell configuration

In this section we consider the other method of describing pattern dynamics that consist of rather long data strings at each time step. These data are observed in many fields of science and/or engineering. For instance, two-dimensional video images are often discretized by an appropriate sam-

TABLE I. Totalistic rule in CA. If one employs M-neighboring cells as G when updating patterns, then $M+1$ cases can occur in $\Sigma_{j\in G}a_j^t$, and for each case the next state can take either 0 or 1.

$\sum_{j\in G} a_j^T$	0	1	!!···!!	M
$f\left(\sum_{j\in G} a_j^t\right)$	0 or 1	0 or 1	···	0 or 1

pling in the time domain and one sampled frame image is decomposed into one-dimensional image signals by raster scanning. Furthermore, if image signals are digitized by sampling in the scanning time domain and amplitude quantization, then the final data are a sequence of long bit patterns. Long binary data strings are, however, difficult to describe and reproduce by 1-2-3 CA because they vary widely in time development of various local patterns, which could give contradictory updating rules when described with 1-2-3 CA, even if the rules are repeatedly applied. Thus, apart from 1-2-3 CA used for the sound data case, we would like to propose a new alternative method to reproduce arbitrarily given binary data strings that also use the other CA rule. The essential point is to introduce totalistic rules with many neighbors. This is a strong contrast to a recurrent neural network model (NN) which enables us to embed given binary patterns as multiple attractors in the form of fixed points or limit cycles, i.e., patterns are recorded in an extended state of real numbers or, in the other words, real values of connection strength between neurons, which are called "synaptic connection strength." This contrasting point was discussed by Nagai *et al.*, who also proved the dynamical equivalency between CA and NN [17,18]. In comparison with NN, the description by CA is quite simple but has to sacrifice retrieval performance, namely "association ability or noise robustness." Rules of CA determine the state of each cell at the next time step depending on the present state of neighboring cells, so that even a one-bit difference between them could

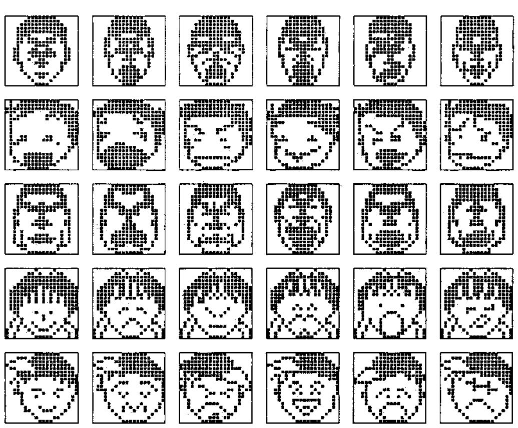

FIG. 6. Pattern sequences that are used in this paper to be reproduced perfectly by certain totalistic rules proposed in this paper. The patterns with strong structures have been used in the works of one of the authors [12] who has extensively studied complex dynamics of NN, including chaos.

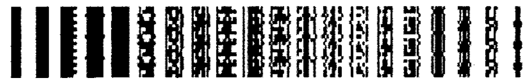

FIG. 7. Periodic pattern dynamics when $20\times20=400$ pixels are reconfigured into one string by raster scanning, where the uppermost periodic cycle is shown in this figure.

lead to totally different states from the desired perfect pattern sequence. We will discuss the noise robustness of our method later in this section, but let us first explain our new method in a little more detail.

The point is *to employ an optimized totalistic rule for each cell*. A totalistic rule is a rule for deciding the state of a cell at the next time step, depending on the number of cells that have the state 1 (or 0) in certain specified neighboring cells where the configuration of neighboring cells can be chosen, if one wants, to optimize or realize the errorless description of data, as shown in the later sections. In this scheme, the totalistic rule of this cell is written as

$$a_i^{t+1}=f_i\left(\sum_{j\in G_i(M)} a_j^t\right), \qquad (3)$$

with respect to the ith cell. Here, $G_i(M)$ represents a set of cells with a given spatial configuration including M cells (M: natural integer number), and f_i is the function that takes 0 or 1 depending on the value $\Sigma_{j\in G_i(M)}a_j^t$. Thus one can know the total number of cases to determine the values of f_i are 2^{M+1}. This is illustrated in Table I. Now, let us show an example that explains the conditions employed in this paper in a little more detail. We prepared the 30 patterns shown in Fig. 6, with each pattern consisting of 400 pixels.

We intend to reproduce the periodic state sequences consisting of five cycles, each of which contains the sequence of the six patterns shown in each row of Fig. 6. This means that once any of the patterns are given, the next pattern in the cycle should be reproduced by applying the specified rule(s), so that one has to find the rule(s) to describe the $6\times5=30$ cases of a two-pattern step. As stated above, each pattern is represented by a 400-bit (cell) string with an appropriate raster scanning. Figure 7 shows the periodic bit pattern sequence for the first cycle in Fig. 6.

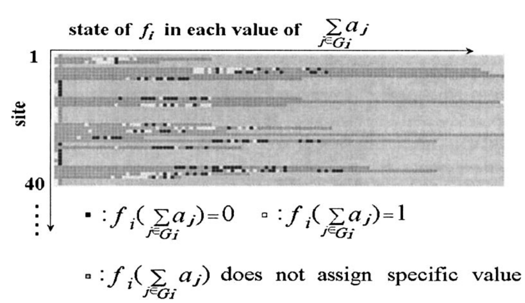

FIG. 8. The totalistic rule for each cell used in perfect reproduction of pattern sequences shown in Fig. 6. The rules for only 40 initial pixels are shown. White cells mean that one should take 1 if $\Sigma_{j\in G}a_j^t$ falls into this case and black cells, and should take 0 if it falls into this case. Gray cells mean that in this description, no cases of reproducing the given pattern sequences shown in Fig. 6 occur.

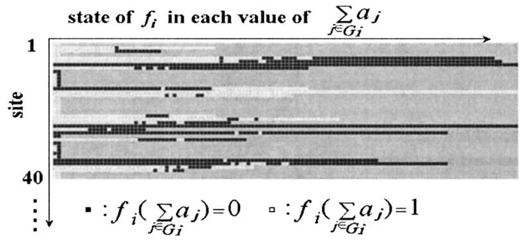

FIG. 9. The same totalistic rule as shown in Fig. 8, where the cells shown in gray are all interpolated to be 1 or 0.

These cycle patterns were chosen to compare dynamical properties occurring in NN, which were intensively investigated in Refs. [10–12]. Now, for each cell, one should search a specified totalistic rule by the following algorithm.

(i) Start from one cell and decide the neighboring number and the configuration of neighbor cells, for instance, the neighbor number is 3 and the configuration is taken as the cell itself and the neighboring cells on both the right and left sides. In taking a symmetric neighboring configuration, for instance r neighbors on both the right and left sides, one has $R=2r+1$ neighboring cells and has to sum up the $2r+1$ cell states for the 30 cases of the given pattern steps.

(ii) To specify the totalistic rule which reproduces the 30 cases of two-pattern steps without error.

(iii) If errorless reproduction is not possible, then increase the number of neighboring cells and, if one wants to optimize the method, change the configuration, continuing until the errorless reproduction is obtained.

(iv) The procedure from (i) to (iv) should be done for all cells, which in the present simulation is 400 cells.

Figures 8 and 9 shows one of the rules determined by the above algorithm with a symmetric configuration of cells. In the figure it should be noted that most of the $2r+1$ cases are not used because the total summation of cell states under a given number and configuration of neighboring cells in the 30 patterns does not exhaust all cases, due to certain pattern structures as observed in Fig. 6. The results of computer experiments are shown in Table II (the symmetric case).

Now, let us investigate the robustness of our method to noise. An important point is that if noise is introduced in an initial pattern or during updating of patterns, unused cases of $\Sigma_{j\in G}a_j$ in perfect pattern sequences occur due to the existence of noise. Thus, one needs to determine the rules, which states whether 1 or 0 should be employed in those cases. For instance, there are $R+1=(2r+1)+1$ cases of $\Sigma_{j\in G}a_j$ for determining each cell state at the next step, where most cases do not occur. There would be many choices because a large number of cases in $R+1$ are not used. In the present paper,

TABLE II. Maximum, minimum, and averaged neighboring numbers in an errorless description with totalistic rules for each cell for the multiple cyclic-pattern sequences shown in Fig. 6.

	Max	Min	Average
Method 1	627	1	178.06
Method 2	129	1	52.19

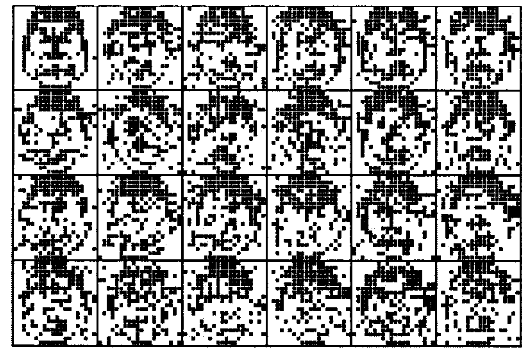

FIG. 10. Updated patterns that start from an initial condition that includes noise (five pixels inverted). It should be noted that the noise considerably deforms the updated pattern from the original ones. However, they are not random patterns and appear to be itinerant in the spaces near the original patterns.

we employ the following interpolated extension of totalistic rules. We propose an idea that used cases (1 or 0 at certain specified cells) should be extended to the unused cases until they are fully interpolated, then any of the $R+1$ cases in $\Sigma_{j\in G}a_j$ can be used, even in the existence of noise during updating. Figure 10 shows an initial pattern with five pixels inverted from $+1(0)\rightarrow 0(+1)$. One can observe that passing time results in differences to the original sequence of patterns, but similar patterns are produced. It should be noted that five-pixels inversion leads not to random patterns but to itinerant orbits that are quite close to the cycles we wanted to describe. This simulation result suggests that the rules could derive new dynamics without the complex mechanism that is necessary for, say, neurodynamics. However, a large deviation from the original patterns may occur, thus this method would be a weak neural network model with respect to noise removal.

B. Optimized errorless description by asymmetric neighboring cell configuration

The problem with the method employed in the previous section is that one needs considerably large numbers of

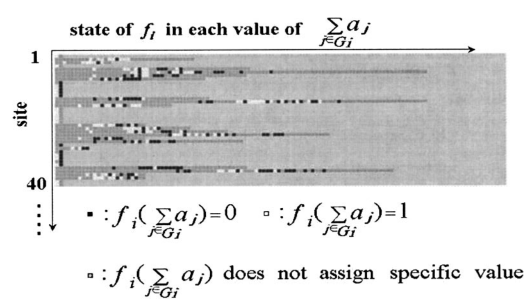

FIG. 11. The totalistic rule used to perfectly reproduce pattern sequences shown in Fig. 6, where the rules are optimized with respect to the configuration of cells (G) to evaluate $\Sigma_{j\in G}a_j$. The rules for only the 40 initial pixels are shown. White cells mean that one should take 1 if $\Sigma_{j\in G}a_j^t$ falls into this case, and black cells mean that one should take 0. Gray cells mean that no case of reproducing the given pattern sequences shown in Fig. 6 occurs in this description.

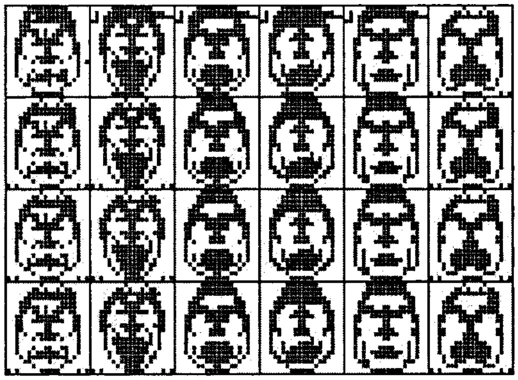

FIG. 12. Updated patterns starting from an initial condition that includes the noise (five pixels inverted), where the initial state is the same as in the case of Fig. 10. It should be noted that the noise does not extend and the updated patterns are almost the same as those we want to reproduce.

neighboring cells to obtain a perfectly reproducible description. To improve this difficulty, we propose to optimize the configuration of neighboring cells. That is, while in the symmetric case, one takes $2r+1$ neighboring cells as the cell itself and r cells on both the right and left sides, in our improved method, one can take $2r+1$ cell configuration asymmetrically, namely an idea to make the configuration adjustable. When we introduced this optimization procedure in the proposed algorithm, great improvement was obtained, as shown in Table II as Method 2. Figure 11 shows an example of optimized cases, where only the rules for the 40 initial pixels are shown. The same experiments are done for the other patterns, which are other face patterns and random patterns, to prove that our improved method is generically effective.

How is the noise robustness of this method improved? The computer experiment shown in Fig. 12 represents that noise robustness is also improved.

It should be noted that, in this figure, we did not employ the interpolation method for unused cases of totalistic rules and, if $\Sigma_{j\in G} a_j$ takes a value that is not assigned, we use a method in which such cells keep the states in the next time step. To evaluate the effect of our interpolation method, we now employ the same method stated in the previous section. The interpolated totalistic rule is shown in Fig. 13, while the results of the computer experiments are shown in Fig. 14,

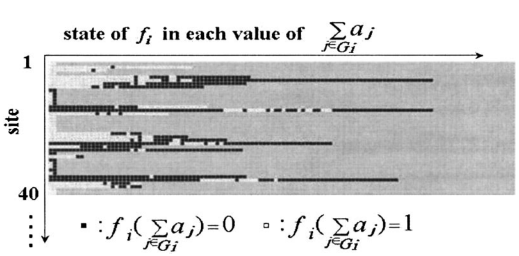

FIG. 13. The totalistic rule in the optimized case, where the unused cases are all interpolated to be 1 or 0. Only the rules for the 40 initial pixels are shown. White cells mean that one should take 1 if $\Sigma_{j\in G} a_j^t$ falls into this case, and black cells should take 0.

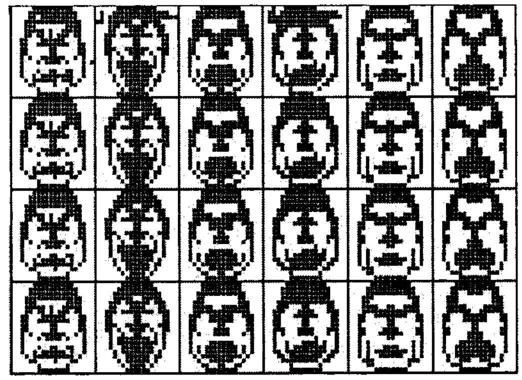

FIG. 14. Updated patterns that start from an initial condition that includes the noise (five pixels inverted), where the initial state is the same as the cases of Figs. 10 and 12. It should be noted that the noise vanishes, meaning that the rule works as a generator of attractor dynamics.

where the noise number and the configuration are the same as in the previous cases.

One can observe that the updating of patterns converges to the original pattern sequence. This means that this interpolation method generates an attractor dynamics.

C. Noise robustness and transitions between attractor patterns due to rule sequences

In this section we have evaluated the noise robustness of dynamics described by subsequent application of the encoded sequence of rules via further computer experiments. In processing numerical calculations, we have found a new dynamical property, i.e., there are transitions between different attractor patterns depending on the number and the positions of bit noises. An example is shown in Fig. 15, where an initial pattern, including a considerable number of randomly configured bit noises, does not make the initial patterns recover the pattern before adding noise, but converges into the cyclic patterns with slightly different bits or into the one of different perfect cyclic patterns. To investigate these dynamical properties more accurately, the following statistical data are evaluated by numerical calculations, i.e.:

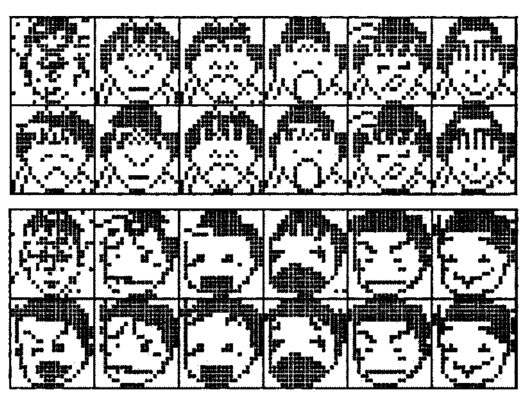

FIG. 15. The two examples in which an initial pattern, including a certain amount of noise, does not recover the pattern before adding noise but converges into the cyclic patterns with slightly different bits (upper) or into one of perfect cyclic patterns (lower).

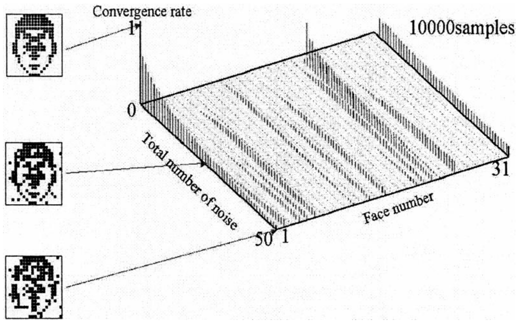

FIG. 16. Statistical data for the 10 000 initial patterns including noise bits from 2–50. Note that the units and the scales are the same as for the previous figures.

(1) To prepare the 10 000 initial patterns including *K*-noise bits ($K=2-50$) with respect to one of the original perfect patterns.

(2) To subsequently apply the 10 000 initial patterns to the encoded sequence of rules for sufficiently long time steps and to confirm that the passing time has converged into cyclic sequences.

(3) To specify the converged perfect periodic patterns to which they belong in Fig. 6 and to calculate the normalized distribution with respect to the 10 000 samples.

An example is shown in Fig. 16. One can observe that the rate of initial patterns which recover the perfect original patterns decreases and the other initial patterns jump to the other perfect cycles as noise increases. This indicates that the encoded sequence of rules is not very robust to noise, but does generate a new dynamical property. This pattern transition is rather generic because we have evaluated the same quantities for all of the encoded sequences, and several results are shown in Fig. 17. These results relate to chaotic dynamics shown by one of the authors in Ref. [12].

IV. CONCLUDING REMARKS

Let us state a summary of our work.

(1) Rule dynamics obtained in perfectly reproduced digital sound signals by 1-2-3 CA make attractor dynamics, in which any initial pattern consisting of 16 bits is pulled into the original pattern dynamics.

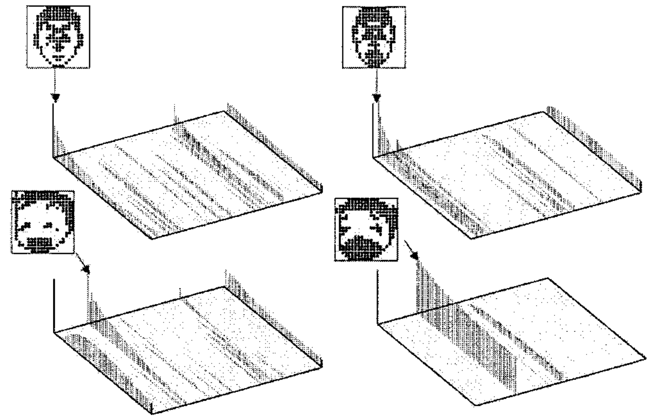

FIG. 17. Cases of evaluating attractor transitions related to other perfect patterns by adding noise bits from 2–50.

(2) Rule dynamics consist of sequences of only two rules in 1-2-3 CA; these two rules belong to the two classes named by Wolfram, Class2 and Class3, namely the class giving a limit cycle and the class giving a chaotic state. This means that a sound signal is standing on the delicate point between ‘‘convergence’’ and ‘‘divergence’’ dynamics.

(3) Perfect description of arbitrarily given pattern dynamics by CA rules has a great advantage in that it needs only a small amount of information to describe and record.

(4) Although this coding is rather weak for noise, it may present other possibilities, such as producing new information for new functions such as memory synthesis, and so on.

ACKNOWLEDGMENTS

The authors would like to thank Professor Y. Aizawa and Professor Y. Nagai for valuable discussions in the early stages of this work. This work has been supported partly by a Grant-in-Aid for Promotion of Science No. 13831008 from the Japan Society for the Promotion of Science and also partly supported by CREST (Core Research for Evolutional Science and Technology) from the Japan Science and Technology Corporation (JST).

[1] R. Shaw, Z. Naturforsch. A **36**, 80 (1981).
[2] T. S. Parker and L. O. Chua, Proc. IEEE **75**, 982 (1987).
[3] K. Ikeda, K. Otsuka, and K. Matsumoto, Prog. Theor. Phys. Suppl. **99**, 295 (1989).
[4] E. A. Jackson, *Perspectives of Nonlinear Dynamics 1 and 2* (Cambridge University Press, Cambridge, 1991).
[5] I. Tsuda, World Future **32**, 167 (1991); also see K. Kaneko and I. Tsuda, *Chaos Scenarios of Complex Systems* (Asakura Publ., Tokyo, 1996) (in Japanese) [the English version is *Complex Systems: Chaos and Beyond* (Springer, Berlin, 2001)].
[6] Hiroshi Fujii, Hiroshi Itoh, Natsuhiro Ichinose, and Minoru Tsukada, Neural Networks **9**, 1303 (1996).
[7] Y. Aizawa and I. Nishikawa, *Dynamical Systems and Nonlinear Oscillators*, edited by G. Ikegami (World Scientific, Singapore, 1986), pp. 210–222.
[8] Y. Aizawa and Y. Nagai, Bussei Kenkyu **48**, 316 (1987) (private communications in Japanese).
[9] S. Kim and Y. Aizawa, Prog. Theor. Phys. **104**, 289 (2000).
[10] P. Davis and S. Nara, ‘‘*Chaos and Neural Networks*,’’ Proceedings of the First Symposium on Nonlinear Theory and Its Application, Hawaii, 1990 (IEICE, Tokyo, 1990), p. 97.
[11] S. Nara and P. Davis, Prog. Theor. Phys. **88**, 845–855 (1992).
[12] S. Nara and P. Davis, Int. J. Bifurcation Chaos Appl. Sci. Eng. **5**, 1205 (1997).
[13] I. Tokuda, T. Nagashima, and K. Aihara, Neural Networks **10**, 1673 (1997).
[14] S. Nara, M. Wada, N. Abe, and J. Kuroiwa, Int. J. Bifurcation Chaos Appl. Sci. Eng. **9**, 1211 (1999).

[15] M. Wada, J. Kuroiwa, and S. Nara, *Proceedings of 1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)* (IEEE, Tokyo, 1999), Vol. 1, pp. 278–283.
[16] M. Wada, J. Kuroiwa, and S. Nara, Phys. Lett. A **306**, 110 (2002).
[17] Y. Nagai (private communications).
[18] A. N. Kolmogorov and V. A. Uspensky, in *Proceedings of the 1st World Congress of the Bernoulli Society*, edited by Y. A. Prohorov and V. V. Sazonov (VNU Science Press, Utrecht, 1984), pp. 3–53.
[19] S. Wolfram, *Theory and Application of Cellular Automata* (World Scientific Singapore, 1986).