Neighborhood Detection Using Mutual Information for the Identification of Cellular Automata

Y. Zhao and S. A. Billings

Abstract—Extracting the rules from spatio–temporal patterns generated by the evolution of cellular automata (CA) usually requires a priori information about the observed system, but in many applications little information will be known about the pattern. This paper introduces a new neighborhood detection algorithm which can determine the range of the neighborhood without any knowledge of the system by introducing a criterion based on mutual information (and an indication of over-estimation). A coarse-to-fine identification routine is then proposed to determine the CA rule from the observed pattern. Examples, including data from a real experiment, are employed to evaluate the new algorithm.

Index Terms—Cellular Automata, identification, mutual information.

I. INTRODUCTION

Cellular automata (CA) are a class of spatially and temporally discrete mathematical systems characterized by local interactions. Because of the simple mathematical constructs and distinguishing features, CA have been widely used to model aspects of advanced computation, evolutionary computation, and for simulating a wide variety of complex systems in the real world [1], [17], [18] and [21].

In many applications, the resulting CA pattern can be observed but the underlying CA rule is unknown. This could be true for example when dealing with some natural systems. The key problem the observer faces is to understand how the system works, this involves identifying the rule and then using the identified model of the system to predict the output. The theory of how CA rules can be extracted from observed patterns of spatio–temporal behavior is therefore fundamental to the study of CA. Essentially, this is an inverse problem, which means that the order of cause and effect is reversed: the observer knows the effects instead of the causes and tries to deduce the CA rule. Often solving inverse problems is difficult because the problem itself can be ill posed. Because of these difficulties determining how the transition rules can be extracted from observed patterns of spatio–temporal behavior has attracted few investigations, but if this problem can be solved, many applications may benefit from it.

Adamatzky presented a sequential and parallel algorithm to determine the local CA transition table [2], and introduced a genetic programming solution with automatically defined functions, to evolve a rule for the majority classification task for one-dimensional (1-D) CAs [3]. However, because most CA use either the von Neumann, the Moore or larger neighborhoods in order to model systems with long-range interactions, the number of potential rules can be very large and the application of the above algorithms can become very complicated and time consuming. For example, a three-site 1-D CA will have \(2^{2^3} = 256\) possible rules while the number of possible rules will explode to \(2^{2^5} = 1.3\times10^5\) for a nine-site 1-D CA. It is therefore often very difficult to scan all the possible rules even using modern computers. To simplify the problem, Yang and Billings [4], [5] showed that CA binary rules can be represented by a simple polynomial model. Using this important observation, the number of parameters to be identified can be substantially reduced, for example, from \(1.3\times10^5\) to 512. Based on the polynomial model form, the identification of CA can be divided into two parts: neighborhood detection and parameter estimation. It is often straightforward to estimate the unknown parameters once the correct neighborhood has been determined, and therefore in this paper we will concentrate on the neighborhood detection problem.

In this paper a new neighborhood detection algorithm is introduced based on mutual information (MI) to provide an initial indication of the temporal and spatial range in the identification of CA. This initial neighborhood is then used to prime a cellular automata-orthogonal least squares (CA-OLS) algorithm to find the correct model terms and unknown parameters in a CA model. This provides a coarse-to-fine identification approach for CA where the MI is used to significantly reduce the potential neighborhood choices which are then optimized using the CA-OLS identification algorithm.

The paper is organized as follows. The structure of CA and an overview of previous investigations are introduced in Section II. The new neighborhood detection algorithm based on MI is proposed in Section III together with a coarse-to-fine identification routine for CA. Three examples, including two simulation studies and a real system, are demonstrated in Section IV, and conclusions are given in Section V.

II. CELLULAR AUTOMATA

A CA is composed of three parts: a neighborhood, a local transition rule and a discrete lattice structure. The local transition rule updates all cells synchronously by assigning to each cell, at a given step, a value that depends only on the neighborhood. Relatively simple binary CA rules can produce highly complex patterns of behaviour, although only two values can be taken in each cell. Because of the advantages, binary CA have attracted many authors to investigate the properties of this class of systems. In the present paper, only binary CA will be considered.

Based on the properties of the transition rule, CA can be classified into two types: deterministic cellular automata (DCA) and probabilistic cellular automata (PCA). The transition rules for DCA are deterministic while those for PCA are statistical because some state flipping of cell values occurs during the evolution of the PCA. Due to the presence of noise, which causes state switching of cell entries during the CA evolution, it is more difficult to identify the rule of PCA than that of DCA. Therefore, the identification of PCA is often investigated independently from the identification of DCA.

A. CA Neighborhoods

The neighborhood of a cell is the set of cells in both spatial and temporal dimensions that are directly involved in the evolution of the cell. Sometimes this includes the cell itself. The neighborhood structure varies depending on the construction of the CA. Consider a 1-D 3-site CA for example where the cell at position \(j\) and at time step \(t\) is denoted as \(c(j,t)\). The neighborhood of cell \(c(j,t)\) could then be a von Neumann neighborhood or other exotic neighborhoods, some of which are shown in Fig. 1 where the black cells denote the cells \(c(j,t)\) and the gray cells denote the neighborhoods. The Exotic 1 neighborhood shown in Fig. 1 encompasses cells from the same temporal scale but different spatial scale than the cells in the von Neumann neighborhood, while the neighborhood of Exotic 2 involves cells from two temporal scales. There are many possible neighborhood structures for a two-dimensional (2-D) CA. The most common structures are the von
The symbol $\lor$ represents the unknown integer parameters model, and the identification of the CA rule is translated into determining the structure of this model and estimating the unknown parameters.

### C. Previous Studies on Neighborhood Detection

In most former studies, the neighborhood was manually predefined as the cells that were immediately close to the cell to be updated. Richards directly selected the Moore structure as the neighborhood of the pattern generated by dendritic solidification [8]. Adamazky set a minimal neighborhood before the identification of a 1-D CA [2]. But for most systems, especially higher order CA, it will often be very difficult to manually choose a candidate neighborhood that just covers the exact neighborhood and rejects the many possible redundant cells. The potential combinatorial possibilities for the neighborhood are large and neighborhood detection is a key step in CA identification.

Based on the polynomial (2), Yang and Billings proposed a modified orthogonal least squares algorithm, abbreviated as CA-OLS, to detect the neighborhood and estimate the parameters in CA models [4], [5]. The preliminary step in this algorithm involves choosing a candidate neighborhood, which can be coarse but must include all correct neighborhoods. Consider a 1-D CA for example and assume the neighborhood of the cell $c(j; t)$ is chosen as $\{c(j - 1; t - 1), c(j; t - 1), c(j + 1; t - 1)\}$; a polynomial model, expressed as (3), can be generated according to (2)

$$c(j; t) = \theta_0 + \theta_1 c(j - 1; t - 1) + \theta_2 c(j; t - 1)$$
$$+ \theta_3 c(j + 1; t - 1) + \theta_4 c(j - 1; t - 1)c(j; t - 1)$$
$$+ \theta_5 c(j - 1; t - 1)c(j + 1; t - 1)$$
$$+ \theta_6 c(j; t - 1)c(j + 1; t - 1)$$
$$+ \theta_7 c(j - 1; t - 1)c(j; t - 1)c(j + 1; t - 1).$$

Determining which terms are significant and which terms are redundant can be derived directly from the error reduction ratio (ERR), which measures the contribution of each candidate term to the updated cell and which is part of the CA-OLS routine. Using the ERR, the candidate terms can be ranked in order of importance and the insignificant terms can then be discarded.

Mei and Billings [9] recently proposed a new neighborhood detection routine, which can refine the candidate neighborhood according to a statistic associated with each combination of candidate neighborhood cells and the updated cell, from which an exact neighborhood can be obtained.

### III. Neighbourhood Detection Based on Mutual Information

Most previous algorithms for the identification of CA, involve the preliminary step of either guessing the candidate neighborhood or performing an initial determination of the candidate neighborhood. The candidate set must be large enough to cover all potential neighborhoods, but too large a range will often involve massive computational time in the subsequent identification. There is a clear tradeoff between computational time and productivity of the model. Hence, the determination of the candidate neighborhood becomes crucial and a failure to select an appropriate neighborhood will often produce an incorrect CA model.

In this paper, a new neighborhood detection approach based on MI is introduced to detect a coarse range for the CA neighborhood without any \textit{a priori} neighborhood information regarding the observed system.

### A. Mutual Information

MI, which was initially proposed by Shannon in 1948 [10], is a measurement of the dependence between two variables. If the two variables are independent, the MI between them is zero. If the two variables are strongly dependent, then the MI between them will be close to 1. MI can measure the state predictability or the memory of a system, represented by a sequence of certain symbols.
Let $x$ be the random variable uniformly chosen from $\{e_1, \ldots, e_m\}$ and let $y$ be the random variable chosen from $\{g_1, \ldots, g_n\}$. Then, the MI of $x$ and $y$, written $I(x; y)$, can be defined by

$$I(x; y) = \sum_{i=1}^{m} \sum_{j=1}^{n} p_{x_i, y_j} \log_2 \frac{p_{x_i, y_j}}{p_{x_i} p_{y_j}}$$

where $s_{x_i, y_j}$ is the counter when $x = e_i$, $s_{x_i}$ is the counter when $y = g_i$, and $s_{x_i, y_j}$ is the counter when $(x = e_i) \cap (y = g_i)$, and $N$ denotes the number of pairs $(x, y)$. The probability values of $p_{x_i, y_j}$ and $p_{x_i, y_j}$ will be more reliable if a larger $N$ is used, and the selection of $N$ will be discussed below.

B. A New Neighborhood Detection Approach Using Mutual Information

Consider the 1-D CA case to illustrate the approach and assume the neighborhood of the cell $c(j; t)$ is $\{c(j-a_1; t-b_1), \ldots, c(j-a_n; t-b_n)\}$. The aim is to determine the maximal spatial lag $a_n$ and the maximal temporal lag $b_n$.

**Definition 1:** A case is defined as a pair of $\{f(R[j; t]), c(j; t)\}$, where $R[j; t]$ is the neighborhood of a cell $c(j; t)$ at time step $t$, and the $c(j; t)$ is the state value of the cell at time step $t$, and $f(R[j; t]) = c_1 + 2c_2 + \ldots + 2^{m-1}c_m$ assuming $R[j; t] = \{c_1, c_2, \ldots, c_m\}$. For example, if the state value of the updated cell $c(j; t)$ is 1 and the state of its neighborhood $R[j; t]$ is $\{0, 1, 1\}$, the case can be described as [5, 1, 1]. Essentially, $R[j; t]$ represents the input and the $c(j; t)$ represents the output of a nonlinear system. If the candidate neighborhood $R[j; t]$ is large enough to cover all the correct neighborhoods, the MI between $f(R[j; t])$ and $c(j; t)$ should be close to 1. If $R[j; t]$ does not contain all the correct neighborhoods, the MI between $f(R[j; t])$ and $c(j; t)$ will be close to 0. According to this rule, this paper presents a new criteria, which introduces MI as a fitness function to establish a measurement for ranking each candidate neighborhood.

Assume the number of component cells of the candidate neighborhood $R[j; t]$ is $m$ and the number of the sampled cases is $N$, the new criteria can be expressed as

$$F = I - OE = \sum_{i,j} p_{k_i, j} \log_2 \frac{p_{k_i, j}}{p_{k_i} p_{j}} - \frac{2^m}{N}$$

where $k_i \in \{0, \ldots, 2^m - 1\}$ and $I_i \in \{0, 1\}$. In expression (5) $p_{k_i}$ denotes the possibility when $f(R[j; t]) = k_i$, $p_{j}$ denotes the possibility when $c(j; t) = I_i$, and $f(R[j; t])$ is given in from Definition 1.

The MI is over-estimated when the number of measurements $N$ is small compared to the number of possible discrete states $2^m$. The over-estimation $OE$ of the MI can be represented by $2^m/N$ [11].

Consider two candidate neighborhoods $R_1[j; t]$ and $R_2[j; t]$ of the cell $c(j; t)$ for a deterministic CA system. If both of these cover all potential neighborhoods and $R_1[j; t] \subseteq R_2[j; t]$, the following relationship is obtained:

$$I(f(R_1[j; t]); c(j; t)) = I(f(R_2[j; t]); c(j; t))$$

$$F(f(R_1[j; t]); c(j; t)) > F(f(R_2[j; t]); c(j; t))$$

(6)

We use $2^m/N$ as an upper bound for the overestimation of the MI between any two finite sets of data. According to expression (6), the final candidate neighborhood can be selected from the peak value of $F$. For the example above, $R_2[j; t]$ should be discarded because of the smaller value of $F$ compared to that of $R_1[j; t]$. The method should therefore avoid selecting a neighborhood with too large a range which can result in a massive waste of computational time at the later stages of CA identification.

As a statistic, the value of MI $F$ is related to $N$. If $N$ is large enough, the result of neighborhood selection will be the same even if each value of $F$ is different. When $N$ is chosen too small, the result of neighborhood selection will be different because there will be insufficient information to generate the correct result. In our experience, it is recommended that the sampled data cases $N$ is selected larger than $50 \times 2^m$, where $n$ denotes the number of candidate neighborhood cells.

The neighborhood detection procedure can now be summarized as:

**Initialization** Assume that the maximal temporal search depth is $t_{\text{max}}$ and set the initial temporary neighborhood as $R_0[c(j; t-1) = \{c(j; t-1)\}$.

1) Collect the case $\{f(R_0[j; t]), c(j; t)\}$, see Definition 1, and calculate the value of $F$ using (5).

2) Increase the temporal range and set the temporary neighborhood as $R_1[c(j; t)] = \{c(j; t-1), c(j; t-2)\}$, and then collect case $\{f(R_1[j; t]), c(j; t)\}$ and calculate the value of $F$.

3) Repeat step 2 until the temporal range reaches $t_{\text{max}}$.

4) Increase the spatial range and reset the temporal range to $t - 1$, which means setting the temporary neighborhood as $\{c(j; t-1), c(j-1; t-1), c(j+1; t-1)\}$, and then collect the case and calculate the value of $F$.

5) Repeat step 1 to step 3 until a peak value of $F$ appears in the spatial direction. The temporary neighborhood with maximal value of $F$ can be selected as the final result.

This neighborhood detection approach produces a range for the correct neighborhood, which considerably reduces the model search in later steps. However, the algorithm cannot guarantee to provide an exactly correct neighborhood. There are cases when the exact neighborhood will be obtained but this may not always be the case. For example, if the neighborhood of a deterministic pattern is symmetrical in the spatial direction, such as $c(j-1; t-1), c(j; t-1), c(j+1; t-1)$, the proposed approach will be expected to produce the exact correct neighborhood. However, if the neighborhood is asymmetrical in the spatial direction, such as $c(j; t-1), c(j+1; t-1)$, a larger range will be detected. This is perfectly acceptable because essentially all we are trying to do is to use the new MI approach to provide an initial estimate of the temporal and the spatial range. This initial neighborhood will then be used to prime the CA-OLS algorithm and to find the correct model terms and unknown parameters. The aim, therefore, is a coarse-to-fine approach where MI is used to significantly reduce the potential neighborhood choices, which can then be optimized using the CA-OLS identification algorithm.

C. Coarse-to-Fine Method of CA Identification

An important step in the identification of CA is to use the MI neighborhood detection approach to restrict the neighborhood search range by providing an initial candidate neighborhood. The remaining steps are the selection of significant model terms and estimation of the unknown parameters. In this paper, the CA-OLS routine [4] will be used for this final step.

Orthogonal least squares (OLS), which was first proposed by Billings in 1988 [6], is widely used in the identification of nonlinear systems. Billings and Yang adapted OLS to the identification of cellular automata and showed that this algorithm can not only estimate the parameters of an equivalent polynomial CA model but can also select the significant model terms. This algorithm is called the forward regression CA-OLS routine. To reduce the computation time for large observed data sets or large neighborhoods Mei proposed a
Fast CA-OLS [12], which discarded some redundant operations and improved the efficiency considerably. The final estimates, however, are the same as ordinary CA-OLS. Other authors have also introduced modifications to the original OLS routine applied to nonlinear system identification to provide improved performance. One algorithm is the locally regularized orthogonal least squares (LROLS), proposed by Chen [13]. The local regularization enforces model sparsity and avoids over-fitting in the model parameters, so the choice of cutoff value is less critical than for ordinary OLS. Another algorithm, named the Combined LROLS and the D-Optimality algorithm (LROLS+D-Optimality), was also proposed by Chen [14]. In this algorithm, it was no longer necessary to specify the cutoff value and the algorithm can automatically terminate by the introduction of a parameter \( \beta \). However, several iterations have to be performed to search for the optimal value in both the above modifications, so the computational time of LROLS and LROLS+D-Optimality is considerably greater than that of Fast CA-OLS, and much larger than ordinary OLS. To illustrate the difference between these algorithms when applied to CA identification massive simulation examples were tested using the four algorithms and the results showed that Fast CA-OLS is typically ten times faster then the other algorithms and obtains almost the same model with few spurious terms. Comparing the advantages and disadvantages of each method, Fast CA-OLS was selected as the core method to estimate the parameters in the new identification routine for binary CA.

The coarse-to-fine identification procedure for CA can be outlined as follows.

1) Detect the coarse spatial and temporal range of the neighborhood using the MI algorithm.
2) Using the result from step 1 to prime Fast CA-OLS, determine the significant terms using the ERR of each term. The correct neighborhood should be produced in this step.
3) Using the neighborhood from step 2, collect new cases then re-prime the Fast CA-OLS to estimate the unknown parameters in (2).

For a deterministic CA, when the neighborhood of the observed pattern is symmetric in the spatial direction, such as \( c(j-1; t-1), c(j; t-1), c(j+1; t-1) \), the new MI neighborhood detection approach can often obtain the exact correct neighborhood in step 1. If the neighborhood is asymmetric, the detected candidate neighborhood using MI will often be larger than the exact correct neighborhood. In such a case, step 3 can be omitted because both ERR and the correct coefficients can be obtained synchronously in step 2.

While processing a probabilistic CA, for both the asymmetric and symmetric neighborhoods, a larger candidate neighborhood will often be detected using MI and all three steps of the coarse-to-fine procedure must be applied to determine the CA model.

### IV. Example Studies

Three simulation examples are employed in this section to demonstrate the application of the new coarse-to-fine algorithm. To show all the steps of this algorithm, initially a probabilistic 1-D example with the symmetric neighborhood and a deterministic 1-D example on the asymmetric neighborhood are discussed. A 2-D real experimental example will then be discussed to show the application of this algorithm to Belousov-Zhabotinsky reaction data.

#### A. Example 1: Identification of a Probabilistic 1-D 3 Site CA Pattern

Consider the Rule R18 [1] on a 200 \( \times \) 200 lattice with three neighborhoods \( \{c(j-1; t-1), c(j; t-1), c(j+1; t-1)\} \). The rule is shown in Table I. Noise was introduced to all three components by flipping the states with 40% probability during the evolution of the CA. This adds severe degrees of noise which is much more disruptive than simply using additive noise on the final pattern. The generated noisy pattern with random initial conditions and a periodic boundary using the above rule is shown in Fig. 3(a). Fig. 3(b) shows the pattern generated by the same rule using the same initial conditions but with no noise. A comparison of Fig. 3(a) and (b) clearly shows the severity of the noise on this example.

The identification procedure is summarized as follows:

**Step 1:** As the preliminary step of the coarse-to-fine routine, the initial maximal temporal search depth \( t_{\text{max}} \) was set to 2, the new neighborhood detection approach was applied and the values of \( F \) for each candidate neighborhood are shown in Table II.

For convenience, define

\[
\begin{align*}
\tilde{s}_1 &= c(j-1; t-1) \\
\tilde{s}_2 &= c(j; t-1) \\
\tilde{s}_3 &= c(j+1; t-1) \\
\tilde{s}_4 &= c(j-1; t-2) \\
\tilde{s}_5 &= c(j+1; t-2)
\end{align*}
\]

so that from Table II, \( F \) reaches a peak when the candidate neighborhood is

\[
\{ \tilde{s}_1, \tilde{s}_2, \tilde{s}_3, \tilde{s}_4, \tilde{s}_5 \}.
\]  

The results using the new algorithm for a larger temporal search depth are illustrated by Fig. 4(a). Inspection of Fig. 4(a) shows that the final selected candidate neighborhood is same as in (7). The set from (7) is therefore selected as the candidate neighborhood. A comparison with Rule R18 shows that this includes all the original neighborhood cells and provides a minimal symmetric spatial range to use to prime the next stage.

**Step 2:** To determine the exact neighborhood and the CA model, cases were collected from the observed pattern and the Fast CA-OLS was applied to evaluate the contribution of each candidate cell to the updated cell. To save space, only the first 20 terms from the results shown in Table III ranked in order of significance are shown.

![Pattern from Rule R18](image)
Step 3: By collecting cases using the neighborhood in expression (8) and applying Fast CA-OLS again, the coefficients in (2) were estimated and these are shown in Table IV, giving the final polynomial model

\[ c(j; t) = s_1 + s_3 - s_1 s_2 - 2 s_1 s_3 - s_2 s_3 + 2 s_1 s_2 s_3 \]  

(9)

The reconstructed pattern, shown in Fig. 3(c), was generated using the polynomial model (9) with the same initial conditions as the observed pattern. A comparison of Fig. 3(b) with Fig. 3(c) clearly shows that they visually appear to be identical. To make a quantitative comparison, the Hamming Distance [22], which is often used to compare two binary images was employed. The Hamming Distance \( D \) between two binary images \( A \) and \( B \) can be expressed by (10) as

\[ D = \sum_{x,y} |A_{x,y} - B_{x,y}| \]  

(10)

where \( A_{x,y} \) and \( B_{x,y} \) denote the value in the position \((x, y)\) in image \( A \) and \( B \).

The calculated Hamming Distance between Fig. 3(b) and (c) was zero, which clearly indicates the identified polynomial model (9) using the new algorithm is an excellent representation of the observed pattern.

Adamatzky [2] proposed an algorithm for the identification of a 1-D cellular automaton. This method is restricted to identifying only deterministic CA. Given an observed pattern produced by a probabilistic CA, this method cannot detect the correct neighborhood or the minimal scale of the rough neighborhood because of the noise.

The identification method for CA, proposed by Yang and Billing [4], was applied to this data set to compare with the new method. The Hamming Distance between Fig. 3(b) and the reconstructed pattern using the model based on Yang’s method was 17644, or a 44% error over 40 000 cells. This demonstrates the effectiveness of the new MI method in determining the CA neighborhood as an initial step.

### B. Example 2: Identification of a Deterministic 1-D CA Pattern With an Asymmetric Neighborhood

Consider the Totalistic Rule R50 [1] on a 200 × 200 lattice with three neighborhood terms \([c(j; t-1), c(j+1; t-2), c(j-2; t-1)]\). The generated pattern with random initial conditions and a periodic boundary condition using Rule R50 is shown in Fig. 5(a). The identification procedure is summarized as follows.

Step 1: The initial maximal temporal search depth \( t_{\max} \) was set to 2, the new neighborhood detection approach was applied and the values of \( F \) for each candidate neighborhood are illustrated in Fig. 4(b).

For convenience, define

\[ s_1 = c(j-2; t-1) \quad s_2 = c(j-1; t-1) \]
\[ s_3 = c(j-0; t-1) \quad s_4 = c(j+1; t-1) \]
\[ s_5 = c(j+2; t-1) \quad s_6 = c(j-2; t-2) \]
\[ s_7 = c(j-1; t-2) \quad s_8 = c(j+0; t-2) \]
\[ s_9 = c(j+1; t-2) \quad s_{10} = c(j+2; t-2) \]

### Table III

**FIRST 20 TERMS PRODUCED BY FAST CA-OLS FOR EXAMPLE 1**

<table>
<thead>
<tr>
<th>Order</th>
<th>Term</th>
<th>ERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>constant</td>
<td>0.23805</td>
</tr>
<tr>
<td>2</td>
<td>( s_3 )</td>
<td>0.05097</td>
</tr>
<tr>
<td>3</td>
<td>( s_1 \times s_3 )</td>
<td>0.10743</td>
</tr>
<tr>
<td>4</td>
<td>( s_1 )</td>
<td>0.10089</td>
</tr>
<tr>
<td>5</td>
<td>( s_2 \times s_3 )</td>
<td>0.088034</td>
</tr>
<tr>
<td>6</td>
<td>( s_1 \times s_2 \times s_3 )</td>
<td>0.05286</td>
</tr>
<tr>
<td>7</td>
<td>( s_2 )</td>
<td>0.04807</td>
</tr>
<tr>
<td>8</td>
<td>( s_1 \times s_2 )</td>
<td>0.03553</td>
</tr>
<tr>
<td>9</td>
<td>( s_2 \times s_4 \times s_5 )</td>
<td>0.00176</td>
</tr>
<tr>
<td>10</td>
<td>( s_2 \times s_3 \times s_4 \times s_5 )</td>
<td>0.00053</td>
</tr>
<tr>
<td>11</td>
<td>( s_1 \times s_4 )</td>
<td>0.00027</td>
</tr>
<tr>
<td>12</td>
<td>( s_1 \times s_5 )</td>
<td>0.00025</td>
</tr>
<tr>
<td>13</td>
<td>( s_1 \times s_2 \times s_6 )</td>
<td>0.00048</td>
</tr>
<tr>
<td>14</td>
<td>( s_2 \times s_6 )</td>
<td>0.00021</td>
</tr>
<tr>
<td>15</td>
<td>( s_1 \times s_3 \times s_4 )</td>
<td>0.00017</td>
</tr>
<tr>
<td>16</td>
<td>( s_1 \times s_2 \times s_3 \times s_6 )</td>
<td>0.00009</td>
</tr>
<tr>
<td>17</td>
<td>( s_1 \times s_3 \times s_5 )</td>
<td>0.00026</td>
</tr>
<tr>
<td>18</td>
<td>( s_2 \times s_3 \times s_6 )</td>
<td>0.00010</td>
</tr>
<tr>
<td>19</td>
<td>( s_1 \times s_5 \times s_6 )</td>
<td>0.00010</td>
</tr>
<tr>
<td>20</td>
<td>( s_1 \times s_2 \times s_4 )</td>
<td>0.00015</td>
</tr>
</tbody>
</table>

Fig. 4. (a) \( F \) values of each candidate neighborhood when \( t_{\max} \) was set to 3 for Example 1. (b) \( F \) values of each candidate neighborhood when \( t_{\max} \) was set to 2 for Example 2.

Inspection of Table III shows that the first eight ERR terms are significantly larger than the remainder. Extracting the most significant result gives the final neighborhood as

\[ \{ s_1, s_2, s_3 \}. \]

(8)
and Anatol. M. Zhabotinsky in 1951, is a spatio–temporal chemical reaction that starts with blue rings propagating from localized regions on a red background. If the dish is given a gentle shake to break up the rings, spectacular geometries appear.

To capture the image of the experiment, a digital video camera using a USB connection to a PC was employed. Operating at full speed, this camera can record roughly 30 fps and the maximal resolution can reach 1024 by 768 pixels. In this experiment, the resolution was selected as 640 by 480 pixels. To enhance the brightness and to prevent an inverted image of the camera in the dish, back lighting was added to the bottom of the reaction dish.

Several preliminary steps were carried out to capture the data set. Because only two values can be taken in binary CA, each pixel in the images must be converted to black or white. The image from the BZ reaction is mainly composed of two colors: blue and red. We converted the blue or close to blue pixels to black, and the red or close to red pixels to white. Another important aspect when collecting real images is to set up the lattice of cells over the image. For example, one point in a real image can be described by a rectangle with four pixels in a digitized image, or could also be described by a rectangle with nine pixels if a larger magnification is used. In this experiment, we adjusted the lattice size such that the tip velocity of the fastest change in the covered image was roughly one cell per time unit. We averaged over 4 × 4 pixel neighborhoods as a representation of the center cell, to produce a 160 pixels × 160 pixels image. A snapshot of one frame processed by the above procedure is shown in Fig. 6(a). The sampled data was composed of seven sequential frames each of which comprised 160 × 160 pixels. Following the coarse-to-fine procedure, the new neighborhood detection algorithm was applied and the results are shown in Table VI. The results indicate that the candidate neighborhood should be selected as the von Neumann structure (see Fig. 2) with temporal lag \( t = 1 \). Because the von Neumann structure is a minimal neighborhood for 2-D CA, Step 2, to select the significant terms, was omitted and Step 3 was applied directly.

For this example, define

\[
\begin{align*}
s_1 &= c(x; y - 1; t - 1) \\
s_2 &= c(x - 1; y; t - 1) \\
s_3 &= c(x; y; t - 1) \\
s_4 &= c(x + 1; y; t - 1) \\
s_5 &= c(x; y + 1; t - 1)
\end{align*}
\]

The results from the Fast CA-OLS algorithm including the ERR values and the coefficients of each terms are shown in Table VII.
TABLE VII
FINAL TERMS AND COEFFICIENTS OF EXAMPLE 3

<table>
<thead>
<tr>
<th>Order</th>
<th>Term</th>
<th>Coefficient</th>
<th>ERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>s₁</td>
<td>1.000000</td>
<td>0.01250</td>
</tr>
<tr>
<td>2</td>
<td>s₂</td>
<td>1.000000</td>
<td>0.09375</td>
</tr>
<tr>
<td>3</td>
<td>s₂ × s₄</td>
<td>-1.000000</td>
<td>0.03044</td>
</tr>
<tr>
<td>4</td>
<td>s₂ × s₅</td>
<td>-1.000000</td>
<td>0.04367</td>
</tr>
<tr>
<td>5</td>
<td>s₃ × s₅</td>
<td>-1.000000</td>
<td>0.03668</td>
</tr>
<tr>
<td>6</td>
<td>s₁ × s₅</td>
<td>-1.000000</td>
<td>0.01103</td>
</tr>
<tr>
<td>7</td>
<td>s₁ × s₄</td>
<td>-1.000000</td>
<td>0.01241</td>
</tr>
<tr>
<td>8</td>
<td>s₁ × s₃</td>
<td>-1.000000</td>
<td>0.00833</td>
</tr>
<tr>
<td>9</td>
<td>s₁ × s₂</td>
<td>-1.000000</td>
<td>0.01042</td>
</tr>
<tr>
<td>10</td>
<td>s₁ × s₂ × s₃</td>
<td>2.000000</td>
<td>0.50000</td>
</tr>
<tr>
<td>11</td>
<td>s₁ × s₄ × s₅</td>
<td>1.000000</td>
<td>0.01658</td>
</tr>
<tr>
<td>12</td>
<td>s₂ × s₃ × s₄</td>
<td>1.000000</td>
<td>0.01518</td>
</tr>
<tr>
<td>13</td>
<td>s₂ × s₃ × s₅</td>
<td>1.000000</td>
<td>0.03265</td>
</tr>
<tr>
<td>14</td>
<td>s₂ × s₄ × s₅</td>
<td>1.000000</td>
<td>0.03784</td>
</tr>
<tr>
<td>15</td>
<td>s₁ × s₃ × s₅</td>
<td>1.000000</td>
<td>0.00972</td>
</tr>
<tr>
<td>16</td>
<td>s₁ × s₂ × s₅</td>
<td>1.000000</td>
<td>0.01122</td>
</tr>
<tr>
<td>17</td>
<td>s₁ × s₂ × s₄</td>
<td>1.000000</td>
<td>0.00937</td>
</tr>
<tr>
<td>18</td>
<td>s₁ × s₂ × s₃</td>
<td>1.000000</td>
<td>0.00962</td>
</tr>
<tr>
<td>19</td>
<td>s₁ × s₃ × s₄ × s₅</td>
<td>-1.000000</td>
<td>0.00174</td>
</tr>
<tr>
<td>20</td>
<td>s₁ × s₂ × s₃ × s₄</td>
<td>-1.000000</td>
<td>0.00374</td>
</tr>
<tr>
<td>21</td>
<td>s₁ × s₂ × s₃ × s₅</td>
<td>-1.000000</td>
<td>0.00142</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

A new neighborhood detection approach has been introduced which is based on finding a coarse range of the neighborhood initially using MI. This method can detect an exact or larger range of neighborhood than the original even in the presence of noise as shown in the examples. The advantage of the new approach is that it can yield significant improvements in efficiency by not only restricting the search range prior to identification to save computational time, but also involving all potential neighborhoods to guarantee the validity of the generated model. This paper also proposes a coarse-to-fine algorithm for the identification of CA, which decompresses the procedure into three steps. The simulated examples and the real experimental data clearly demonstrate the performance of the new algorithm.

ACKNOWLEDGMENT

The authors are very grateful to A. Routh, Department of Chemical Engineering, University of Sheffield, who set up and conducted the real experiment.

REFERENCES