Pattern Generation Using Likelihood Inference for Cellular Automata

Radu V. Craiu and Thomas C. M. Lee

***Abstract*—Cellular automata are discrete dynamical systems which evolve on a discrete grid. Recent studies have shown that cellular automata with relatively simple rules can produce highly complex patterns. We develop likelihood-based methods for estimating rules of cellular automata aimed at the re-generation of observed regular patterns. Under noisy data, our approach is equivalent to estimating the local map of a stochastic cellular automaton. Direct computations of the maximum likelihood estimates are possible for regular binary patterns. The likelihood formulation of the problem is congenial with the use of the minimum description length principle as a model selection tool. We illustrate our method with a series of examples using binary images.**

***Index Terms*—Binary patterns, cellular automata, maximum likelihood estimation, minimum description length principle, neighborhood selection, rule estimation, stochastic cellular automata.**

I. Introduction

A CELLULAR AUTOMATON (CA) [13]; [14] is a discrete system which evolves in discrete time over a lattice structure composed of a large number of cells. The state of each cell belongs to a finite set and is updated according to a local rule operating on a given neighborhood. More precisely, the CA updates at discrete times the entire lattice and the dynamics of the changes are given by a *local map* which is used to determine the new state of each cell from the current states of certain neighboring cells. The local map is completely determined by two components, the *neighborhood*, which includes all the cells that influence a given cell and the *rule*, which specifies how the neighborhood influences it. The classical CA evolves according to deterministic rules. Even if these rules are relatively simple, the corresponding cellular automata can produce structures with a high level of complexity. Wolfram [15] characterizes one-dimensional cellular automata according to their attractor sets. The high level of complexity shown by some of the cellular automata (e.g., those of Class 4 as described by [17, p. 235]) makes it reasonable to investigate whether CA systems can be used to reproduce a variety of patterns that are met in nature.

Manuscript received April 27, 2004; revised June 28, 2005. The work of R. V. Craiu was supported in part by a Grant from the Natural Sciences and Engineering Research Council of Canada and the work of T. C. M. Lee was supported in part by the National Science Foundation under Grant 0203901. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Attila Kuba.

R. V. Craiu is with the Department of Statistics, University of Toronto, Toronto, ON M5S 3G3 Canada.

T. C. M. Lee is with the Department of Statistics, Colorado State University, Fort Collins, CO 80523 USA.

Digital Object Identifier 10.1109/TIP.2006.873472

Other interesting patterns may also be produced if one employs a nondeterministic CA. A natural extension is the so-called *stochastic cellular automaton (SCA)* in which the local map has a probabilistic component. There is a considerable amount of freedom in the way one chooses to randomize the rule. For instance, Burks [1] chooses at random the rules that are applied in the updates and Ingerson and Buvel [4] allow, for each cell, a certain probability that the respective cell is not updated. Lee *et al.* [7] introduce the adaptive SCA in which the probabilistic rules are nonuniform. The SCA we consider in this paper is equivalent to a deterministic CA on which a certain noise signal has been applied. Here we also assume that the same stochastic rules are applied over the whole system.

The application of CA to pattern generation and recognition can be developed in many directions. A classical reference for pattern recognition with CA is Preston and Duff [9]. In addition, one may try to use the CA to generate patterns with pre-specified properties (e.g., with various properties of randomness as in [8] or [16]). Alternatively, given a certain pattern observed in nature, one may want to construct a local map such that the corresponding CA generates patterns similar to the one observed [10]. In a setup somewhat different to the one here, Yang and Billings [19] use genetic algorithms to recover the local map of a deterministic CA using spatio-temporal patterns produced by it. Although the statistical tools seem to be appropriate in the context of fitting a CA/SCA to a particular set of noisy data, the connections between statistics and CA in the context of rule recovery are, as far as our knowledge, sparse in the literature. One exception is the paper by Turin [12] in which a hidden Markov chain model is fitted via the EM algorithm. However, the procedure is quite cumbersome and no example is given to illustrate its performance.

The objective of this paper is to develop statistical methods for the detection of a local rule (or map) of a SCA and estimation of its parameters. Consider for instance the simple binary CA example illustrated in Figs. 1 and 2. We present there the local rule as well as a series of two successive realizations obtained from the *initial states*. Loosely speaking, these initial states are randomly generated pixel values and their formal definition is given in Section II. In this example, the state of a cell is dependent only on the states of the three closest cells situated in the previous line. It is clear that, given an initial row of cells and by applying the local rule, one can run the CA and produce the next pixel line $(t = 1)$. This process can be repeated to generate a sequence of pixel lines. These pixel lines may be stacked on top of each other and displayed in the form of an image in which each line of pixels represents an update of an CA/SCA that involves only the previous lines. However, here we would like to be able

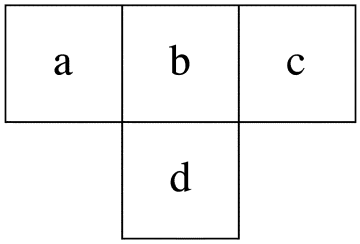


Fig. 1. Neighborhood of size 1×3 for CA/SCA: during the updating processing the future state value of pixel d depends on the state values of pixels (a, b, c). For (deterministic) CA the value of d is uniquely determined by the values of (a, b, c), while for SCA the value of d is a probability function of the values of (a, b, c).

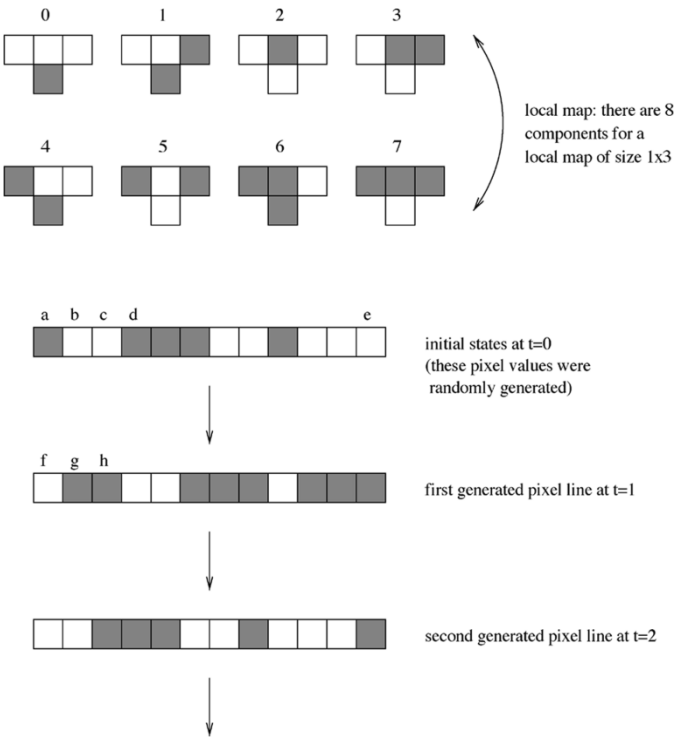


Fig. 2. Binary CA example with rule neighborhood size 1×3. To generate the value of pixel g at $t = 1$, one looks at the initial state values of pixels (a, b, c) at $t = 0$, which are (black, white, white). These three pixel values correspond to component 4 of the local map, which assigns black to pixel g. Similarly, to generate the value for pixel h, one looks at pixels (b, c, d). For pixel f, we use a cylindrical boundary assumption and look at pixels (e, a, b).

to do the reverse. That is, we assume that the local rule is unknown and all that is available is the image formed by the pixel lines. The aim is to reconstruct this unknown local map "hidden" behind the image. By assuming that the underlying generating process is a SCA, we enlarge considerably the number of possible models. For instance, consider the simple rule in which a cell is the opposite of the cell directly above it. If we were to consider only cellular automata it is clear that this simple local map could not produce the sequence of realizations shown in Fig. 2. However, if we allow the probabilistic "contamination" of the rule, then we cannot reject automatically the above possibility and more sophisticated methods are required for rule selection.

One can also apply these methods to a variety of regular images, as it is has been shown recently [17, pp. 232–234] that

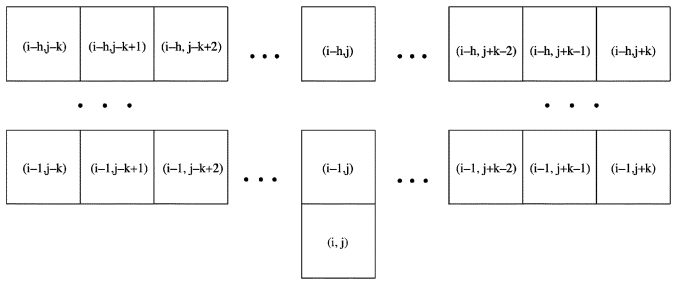


Fig. 3. Neighborhood used in the cellular automata.

CA and SCA can produce very good approximations to a large number of natural patterns.

The approach proposed here for the local map reconstruction relies on a nonparametric model for which one can use likelihood-based methods for estimation and model selection. While the dependence between the neighborhood and the rule can be modeled using a parametric family of distributions, such models are usually too simple to capture the intricate structure of the pattern (e.g., [2]). The patterns considered for reproduction are binary. In the following section, we formally define the CA and SCA, introduce notation and describe the data available. The estimation involves two steps, intrinsically related. First, one needs to estimate the parameters of the stochastic process superimposed over the CA. Second, we must select the local map. Estimation procedures for binary patterns with rules based on medium-sized neighborhoods are described in Section III. The identification of the local map is a model selection problem. For likelihood-based methods developed for signal processing, such as the ones used in this paper, the minimum description length (MDL) principle is a natural tool for model selection. In Section IV, we briefly describe the MDL principle and in Section V we illustrate our methods with a few simulated and real examples.

II. Notation and Data

A deterministic CA is composed of three parts: a discrete two-dimensional lattice $\mathcal{D}$, a neighborhood V and an updating rule for the local map, f. The lattice (i.e., the image) is defined as $\mathcal{D} = \{d_{ij} : 1 \le i \le M, 1 \le j \le N\}$. We will assume here that the lattice $\mathcal{D}$ has at each point a cell in a state that is characterized by the random variable X_{ij} that, in the case of a binary image, takes values in the set $\{0, 1\}$ for all $1 \le i \le M$ and $1 \le j \le N$. A white cell will be assigned the value zero, and a black cell has value one. In this paper we will consider neighborhoods V of size $h \times (2k + 1)$ as shown in Fig. 3, i.e., the state of cell d_{ij}, X_{ij}, is stochastically determined by the values of the cells $d_{i-h\ j-k}$,$d_{i-h\ j-k+1}$,…,$d_{i-h\ j+k}$,…,$d_{i-1\ j-k}$,$d_{i-1\ j-k+1}$, …, $d_{i-1\ j+k}$. Local maps with rules of the type shown in Fig. 3 are inspired by Wolfram [17, Ch. 3 and 6] who showed that many patterns can be generated using a deterministic CA with such rules.

In our applications we consider the following SCA. We assume that the state X_{ij} is a random variable with distribution defined by

$$P\left(X_{ij} = 1 | \tilde{x} = (x_{i-hj-k}, \ldots, x_{i-1j+k})\right) = \theta_{\tilde{x}} \qquad (2.1)$$

where $\tilde{x} = (x_{i-h\ j-k}, \ldots, x_{i-1\ j+k})$ is any of the $2^{h\times(2k+1)}$ possible realizations of the $h \times (2k+1)$-tuple $\tilde{X}_{ij} = (X_{i-h\ j-k}, X_{i-h\ j-k+1}, \ldots, X_{i-h\ j+k}, \ldots, X_{i-1\ j-k}, X_{i-1\ j-k+1}, \ldots, X_{i-1\ j+k})$. An attentive reader will have noticed that for those cells situated close to one of the side edges of the lattice $\mathcal{D}$ the rule given by (2.1) cannot be applied directly. One can account for boundary conditions by allowing a modification of (2.1) in which the evolution of those cells close to the edge depends only on the cells in the neighborhood that are also in the lattice. Evidently, this increases the number of parameters but can be incorporated in a straightforward manner in the present approach. However, to simplify the presentation and the computational load, we will assume that the lattice is in fact a cylinder in which the cells on one lateral edge are adjacent to the cells situated on the opposite edge. Corresponding to a rule given by (2.1), the first h lines of the cylinder are considered the *initial states*. Depending on the situation, these initial states can be obtained in one of the following manners. If, given an observed image, one would like to estimate its unknown rule and use the estimated rule to reproduce further images that mimic the original observed image, these initial states are copied from the corresponding rows in the original image. Otherwise, these initial states can be randomly generated.

To see the connection between the deterministic CA (in which the θs are 0 or 1) and the SCA defined above, imagine that the updates are performed row by row. After an entire row i is updated using the CA the state x_{ij} of each cell d_{ij} is flipped with a small probability $p_{\tilde{x}}$ that may depend on the cells in the neighborhood of d_{ij}. If we assume that the flipping probabilities are constant across rows, the resulting stochastic dynamical system is equivalent to the SCA defined above. It is worth mentioning that one could imagine different noising schemes applied to a deterministic CA so that the resulting discrete dynamical system is equivalent to an SCA with rules similar to (2.1). For example, one could perform a deterministic update of the whole cylinder $\mathcal{D}$ and *at the end* perform a random flip of all the cells. Therefore, the SCA considered in (2.1) covers a wide range of noising processes and is likely to perform well for the generation of regular patterns.

Such a regular pattern will be represented by a cylinder in which each cell is a pixel which takes only two values, 0 and 1. Of interest will be the determination of the neighborhood constants h and k, and the estimation of $\theta_{\tilde{x}}$ for all possible configurations $\tilde{x}$. The parameter space has dimension $2^{h\times(2k+1)}$ so that even for binary images the parameter space rapidly becomes very large when $k, h > 3$. In Sections IV and V below we show, respectively, how $\theta_{\tilde{x}}$ and (h, k) can be estimated using the data observed.

III. Likelihood for Binary Patterns

In a binary image, each cell has only two possible states, 0 or 1. For any value of (h, k), we write $\boldsymbol{\theta}$ as the set of all possible $\theta_{\tilde{x}}$s. The likelihood function $L(\boldsymbol{\theta})$ of $\boldsymbol{\theta}$ is

$$L(\boldsymbol{\theta}) = \prod_{i=h+1}^{M} \prod_{j=1}^{N} \left\{ \theta_{\tilde{x}_{ij}}^{x_{ij}} \left(1 - \theta_{\tilde{x}_{ij}}\right)^{1-x_{ij}} \right\} \tag{3.1}$$

where x_{ij} and $\tilde{x}_{ij}$ are, respectively, the cell value and the neighborhood configuration of cell d_{ij}. It should be noted that, for any fixed value of $\boldsymbol{\theta}$, $L(\boldsymbol{\theta})$ gives the corresponding probability of generating a given pattern $\{x_{ij} : h+1 \le i \le M, 1 \le j \le N\}$. The above expression for $L(\boldsymbol{\theta})$ can be derived by observing that the distribution of each cell state (i) follows a Bernoulli distribution with parameter $\theta_{\tilde{x}_{ij}}$ and (ii) depends only on the *previous h lines*. Thus, the likelihood can be split into a product of conditional probabilities, each of which representing

$$P(X_{ij} = x_{ij} | \tilde{X}_{ij} = \tilde{x}_{ij}) = \left\{ \theta_{\tilde{x}_{ij}}^{x_{ij}} \left(1 - \theta_{\tilde{x}_{ij}}\right)^{1-x_{ij}} \right\}$$

for $h < i \le M$ and $1 \le j \le N$. Notice that, since we assume a cylindrical shape, there are no edge conditions. The top h lines are considered to be the starting values of the CA/SCA. To simplify notation, from now on $\theta_{\tilde{x}}$ will be written as θ_i, where i is the value denoted by the neighborhood configuration $\tilde{x}$ when the configuration is viewed as a binary number. For example, if $h = k = 1$ (i.e., a neighborhood size of 1×3) and $\tilde{x} = (1, 0, 1)$, then i is 5, as 101 is 5 in binary digit representation. Thus, i ranges from 0 to $2^{h\times(2k+1)} - 1$.

A general question of interest in estimation is whether the parameters are identifiable (or estimable) from the data at hand. Amongst others, in the following two situations not all the θ_is can be estimated. First, if the neighborhood size is too large compared to the image size, i.e., $2^{h\times(2k+1)} > MN$, then not all the θ_is can be estimated. It is because in this case there are only MN "data points" while the number of unknown parameters is $2^{h\times(2k+1)}$. The second situation occurs when a rule can be sufficiently represented by a rule with a smaller neighborhood size. One example is when all or almost all θ_is share the same common value, τ, say. In this case the neighborhood size is hard to estimate, as rules with different neighborhood sizes but with the same common value for all θ_is would generate patterns with the same statistical properties. In fact, all these rules label independently each pixel black with probability τ. In what follows, we will exclude those situations in which the parameters are not identifiable, such as the two situations discussed above.

Meanwhile, even if the θs are identifiable, the question of uniqueness is more difficult to answer. Suppose that a given pattern is rotated with a certain angle $\alpha \in (0, 2\pi)$. While the image remains the same, in all but few cases in which the pattern is invariant to rotations, the parameter estimates will be different for different αs. This feature can be used in the estimation process as it may be the case that patterns which are difficult to estimate in their original configuration may be easier to reproduce once a similar transformation is applied.

Another issue of great impact in estimation is the dimension of the parameter vector $\boldsymbol{\theta}$. For small values of k and h say if $k \le 3$ and $h \le 2$, the direct computation of the maximum likelihood estimates can be obtained directly from (3.1). More precisely, for each of the $2^{h\times(2k+1)}$ possible configurations $\tilde{x}$, we can compute $n_i^{(1)}$, the number of times the rule assumed the value 1, and $n_i^{(0)}$ the number of times the rule assumed the value 0. Then, the parameter estimates are

$$\hat{\theta}_i = \frac{n_i^{(1)}}{n_i^{(1)} + n_i^{(0)}}. \tag{3.2}$$

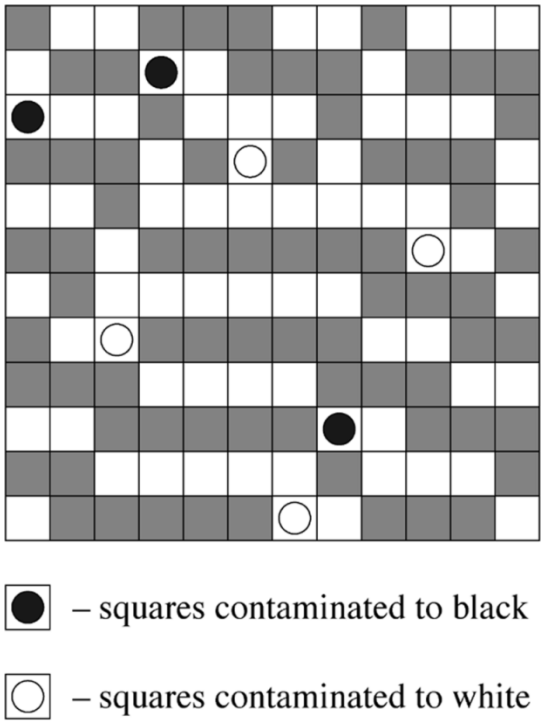


Fig. 4. Binary SCA example with the same local map as in Fig. 2. The first line of pixels is generated at random. With small probability, white pixels are contaminated into black pixels and vice versa. The contaminated pixels are shown using circles inside squares. The color of the circle represents the contamination color.

However, for large values of k and h, the parameter space is too large to be computationally manageable and a different approach is required. Some alternative models for dealing with the high-dimensionality of the space of possible rules are currently under investigation.

TABLE I
8 EXAMPLE RULES AND THE ESTIMATION RESULTS. RECALL THAT THE NUMBER OF REPETITIONS WAS 500

Example Rule	h	$2k+1$	number of θ_i's	number of correct estimates for h and k
1	1	3	8	497
2	1	3	8	500
3	1	5	32	500
4	1	5	32	500
5	2	3	64	500
6	2	3	64	500
7	2	5	1024	500
8	2	5	1024	500

TABLE II
TRUE θ_i AND THE AVERAGED ESTIMATED $\hat{\theta}_i$ FOR EXAMPLE RULES 1 AND 2. NUMBERS IN PARENTHESES ARE THE CORRESPONDING ESTIMATED STANDARD ERRORS FOR THE $\hat{\theta}_i$S. RECALL THAT THE ESTIMATED θ_iS ARE RESTRICTED TO THE RANGE [0.0001, 0.9999]

		Example Rule 1		Example Rule 2	
neighborhood configuration $\tilde{x}$	i	θ_i	$\hat{\theta}_i$	θ_i	$\hat{\theta}_i$
000	0	0.03572	0.03574 (0.01514)	0.9974	0.9974 (0.002127)
001	1	0.0000	0.0001 (0.00000)	0.02908	0.02915 (0.007024)
010	2	0.1166	0.1166 (0.009886)	0.0000	0.0001 (0.00000)
011	3	0.03043	0.03038 (0.007368)	0.0000	0.0001 (0.00000)
100	4	1.0000	0.9999 (0.00000)	0.0000	0.0001 (0.00000)
101	5	1.0000	0.9999 (0.00000)	0.8191	0.8192 (0.01708)
110	6	0.04253	0.04253 (0.008475)	1.0000	0.9999 (0.00000)
111	7	0.9323	0.9322 (0.01935)	1.0000	0.9999 (0.00000)

IV. AUTOMATIC CHOICE OF NEIGHBORHOOD SIZE USING MDL

Arguably, the most important aspect of the estimation of the SCA is the correct identification of the rule's neighborhood. In statistical terms, the problem of selecting the most appropriate k and h is a model selection problem. For each pair (k, h), the parameter space has $2^{h\times(2k+1)}$ parameters so overestimating k or h easily results in overfitting. In turn, this implies that the rules would tend to model the noisy component of the SCA (as well as the deterministic portion). On the other hand, if the values of k and h are too small (i.e., if they are underestimated), the SCA used for reconstruction does not capture all the distinct features of the pattern.

Given a binary image y generated by a SCA with an unknown rule, this section considers the problem of estimating its neighborhood size (i.e., the values of h and k). The minimum description length (MDL) principle will be adopted to tackle this problem. In particular, we shall focus on the so-called *two-part* encoding scheme. Once h and k are determined, maximum likelihood estimates of θ_i can be computed by (3.2). The MDL principle is based on ideas from information theory which were adapted to statistical purposes by Rissanen [11]. It has been successfully applied to tackle many different image processing problems, e.g., see Xie *et al.* [18]. For introductory tutorials on the topic, consult, for example, Hansen and Yu [3] and Lee [5].

In the current context, the MDL principle *defines* the best combination of h and k, or the best neighborhood size, as the one that produces the shortest code length that completely describes the observed binary image y. Here, the code length of an object can be taken as the amount of memory space that is required to store the object. For the so-called two-part MDL, a classical way to store the observed image y is to split y into two parts.

1) A fitted model that summarizes the average (or mean) behavior of many (imaginary) observed images y. Denote the estimates, obtained from y, of h, k and $\boldsymbol{\theta}$ as $\hat{h}$, $\hat{k}$, and $\hat{\boldsymbol{\theta}}$ respectively. Then such a fitted model can be completely specified by the initial states and $\{\hat{h}, \hat{k}, \hat{\boldsymbol{\theta}}\}$.
2) The portion of y that is unexplained by the fitted model. This measures how much the observed image y is deviated from its averaged behavior. Sometimes it is useful to treat this part as the "residual" component of the problem.

Now, the task is to derive expressions for the code length for these two parts so that h and k can be estimated by minimizing the sum of these code length expressions. In below the code

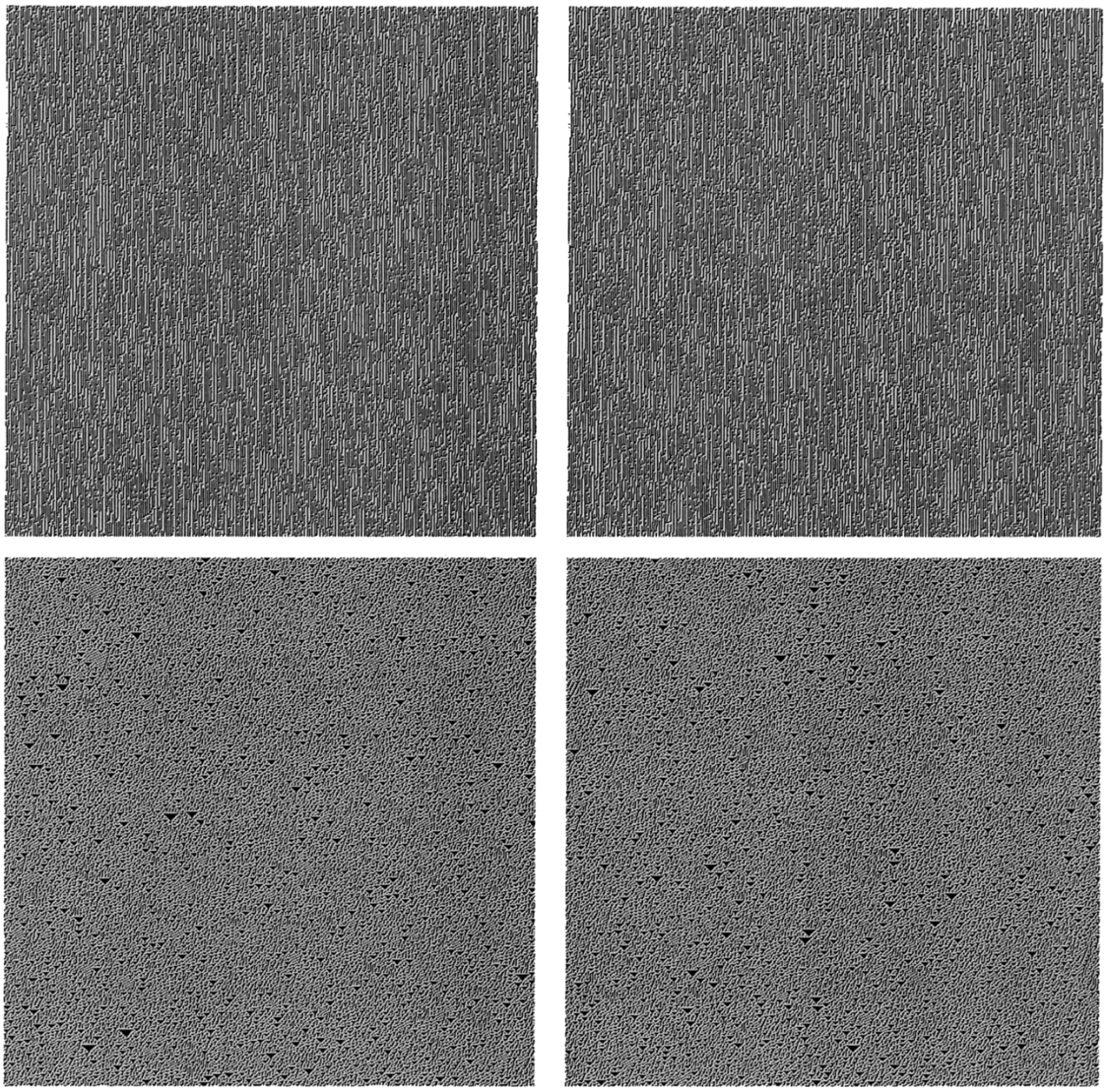


Fig. 5. Left column displays images generated from Example Rules 1 and 2. Displayed in the right column are typical images simulated from rules that are estimated from the corresponding images in the left column.

length of an object z, in terms of number of bits, is denoted as $CL(z)$.

To proceed, we apply the above two-part decomposition to the total code length $CL(y)$ of the observed image y

$$\begin{aligned} CL(y) &= CL(\text{“observed binary image”}) \\ &= CL(\text{“fitted model”}) + CL(\text{“residuals”}) \\ &= CL(\text{“initial states”}) + CL\left(\{\hat{h}, \hat{k}, \hat{\boldsymbol{\theta}}\}\right) \\ &\quad + CL(\text{“residuals”}) \\ &= CL(\text{“initial states”}) + CL(\hat{h}) + CL(\hat{k}) \\ &\quad + \sum_i CL(\hat{\theta}_i) + CL(\text{“residuals”}). \end{aligned} \tag{4.1}$$

Note that MDL defines the “best” combination of h and k as the minimizer of $CL(y)$.

Now, we calculate (4.1) term by term. First since that there are $\hat{h}N$ pixels in the initial states and that each pixel can take on two possible values, we have $CL(\text{“initial states”}) = \log_2 \hat{h}N = \log_2 \hat{h} + \log_2 N$. In below the term $\log_2 N$ will be dropped, as it is a constant with respect to the minimization. Next as both $\hat{h}$ and $\hat{k}$ are integers, their corresponding code lengths $CL(\hat{h})$ and $CL(\hat{k})$ are respectively $\log_2 \hat{h}$ and $\log_2 \hat{k}$ bits. For the third term, the following result of Rissanen [11, pp. 55–56] can be applied: if a (real-valued) parameter estimate is estimated from N data points, then it can be effectively encoded with $(1/2)\log_2 N$ bits. Since, for any given i, $\hat{\theta}_i$ is estimated from $n_i^{(1)} + n_i^{(0)}$ data points, $\sum_i CL(\hat{\theta}_i) = (1/2)\sum_i \log_2(n_i^{(1)} + n_i^{(0)})$. For the last residual term, Rissanen [11] demonstrates that it is equal to the negative of the log (base 2) conditional likelihood function given the fitted model. For the present case this simplifies to

$$\begin{aligned} CL(\text{“residuals”}) &= -\log_2 L(\hat{\boldsymbol{\theta}}) \\ &= -\sum_{i=h+1}^{M}\sum_{j=1}^{N} \log_2 \left\{ \theta_{\tilde{x}_{ij}}^{x_{ij}} \left(1-\theta_{\tilde{x}_{ij}}\right)^{1-x_{ij}} \right\} \\ &= -\sum_{l=1}^{n} \left\{ n_l^{(1)} \log_2 \hat{\theta}_l + n_l^{(0)} \log_2 (1-\hat{\theta}_l) \right\} \end{aligned}$$

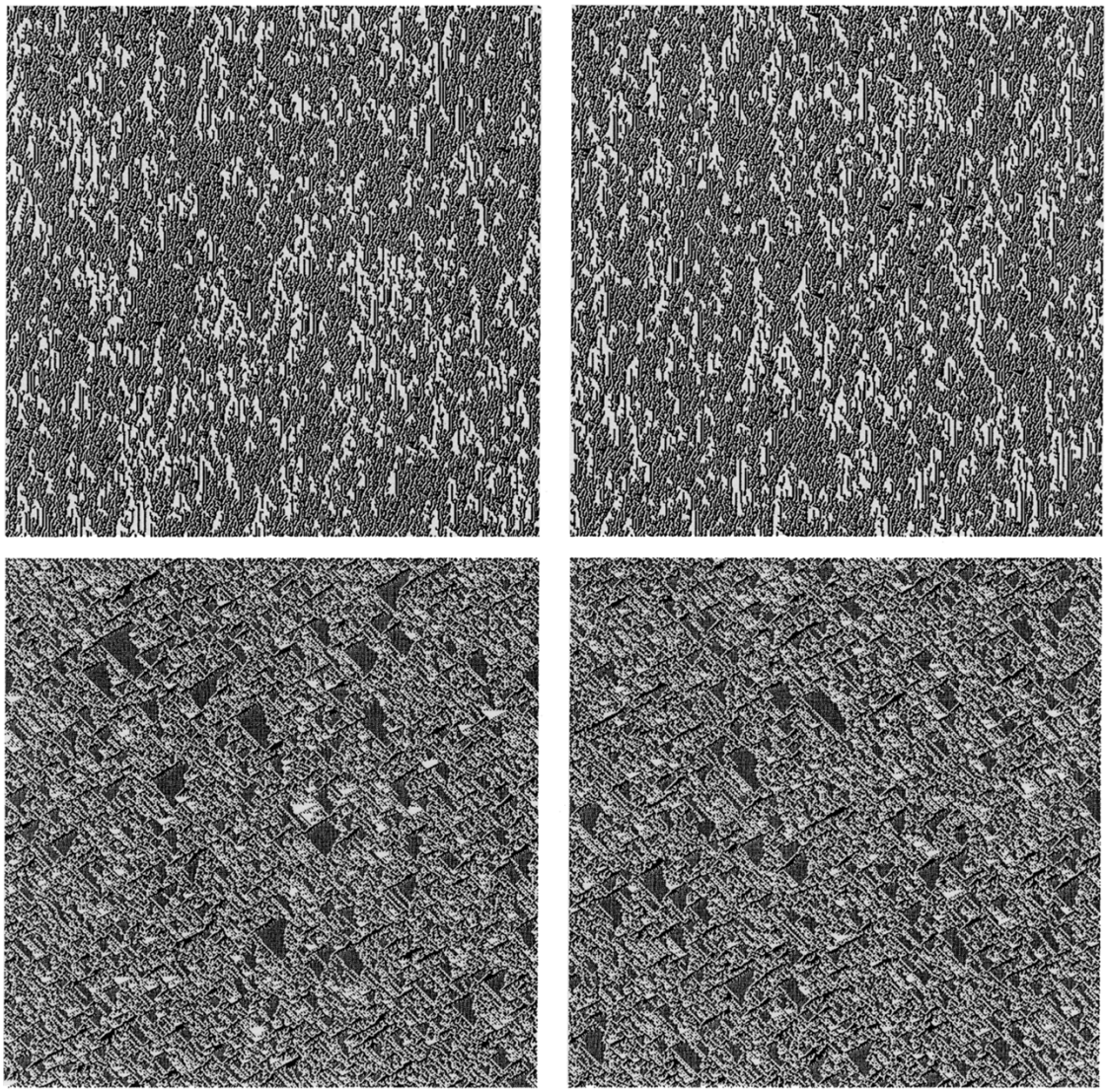


Fig. 6. Similar to Fig. 5, on the left are images generated from Example Rules 3 and 4; on the right are typical images simulated from rules that are estimated from the corresponding images in the left column.

where $n = 2^{\hat{h}\times(2\hat{k}+1)}$. Combining these results and changing $\log_2$ to log, one obtains the following approximation, denoted by $\mathrm{MDL}(\hat{h},\hat{k})$, to $\mathrm{CL}(y)$

$$\mathrm{MDL}(\hat{h},\hat{k}) = 2\log\hat{h} + \log\hat{k} + \frac{1}{2}\sum_{i=1}^{n}\log\left(n_i^{(1)} + n_i^{(0)}\right) - \sum_{i=1}^{n}\left\{n_i^{(1)}\log\hat{\theta}_i + n_i^{(0)}\log(1-\hat{\theta}_i)\right\}. \quad (4.2)$$

Thus the MDL principle suggests estimating h and k as the joint minimizer of $\mathrm{MDL}(\hat{h},\hat{k})$. Once the estimates of h and k are obtained, maximum likelihood estimates of θ_i can be computed by (3.2). For the reason of avoiding numerical instability (e.g., taking the logarithm of 0s), in our implementation estimates for θ_i are restricted to the range $[\epsilon, 1-\epsilon]$, where ϵ is set to 0.0001.

V. Numerical Results

A. Illustrative Example

We start with a simple example meant to illustrate the analysis in detail. Fig. 4 shows a lattice with 12 rows and 12 columns of pixels. The first row of pixels has been generated at random and subsequent rows are updated according to the local map shown in Fig. 2. With probability $\theta = 0.05$ each pixel in a given row will switch its color *before* the rule is applied to that particular row. We refer to such pixels as being contaminated and the new color is called the contamination color. In Fig. 4 the pixels that were contaminated to black are represented with a black disk inside the square while the other contaminated pixels are represented with a circle inside the square. For instance, the pixel situated in second row and the fourth column was white according to component 3 of the local map, but due to contamination it switches to black. Since in practice there is no way of knowing which cells have switched color, we assume that the rule is applied without differentiating between contaminated and noncontaminated pixels.

Suppose now an image such as this 12×12 lattice is observed. The task then is to estimate the neighborhood size (h,k), as well as the corresponding θ_is, of the rule that generated the image. Notice that once estimates for (h,k) are specified, unique maximum likelihood estimates of the

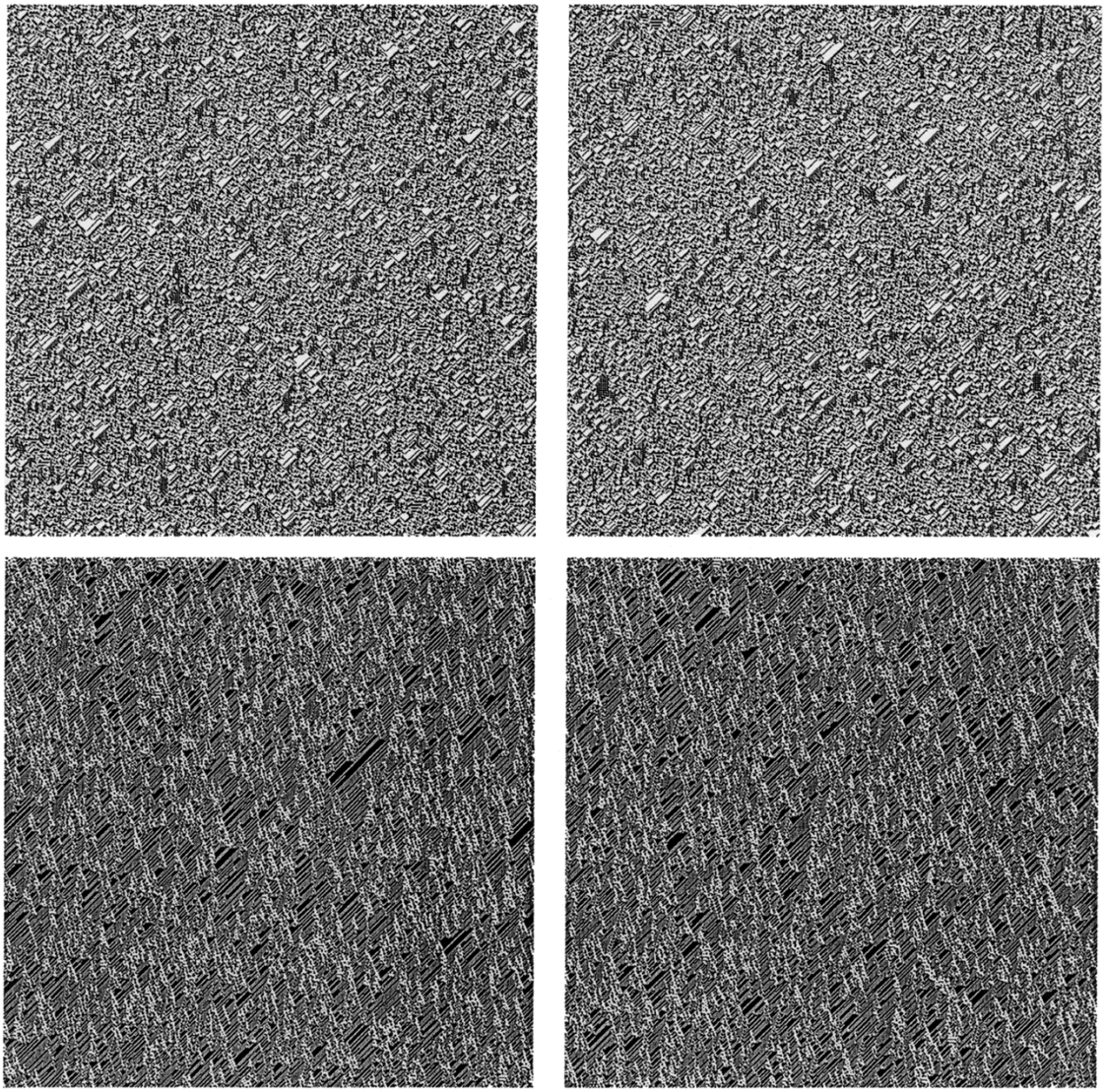


Fig. 7. Similar to Fig. 5, on the left are images generated from Example Rules 5 and 6; on the right are typical images simulated from rules that are estimated from the corresponding images in the left column.

θ_is can be computed using (3.2). As suggested before, the best estimates of (h, k) are given by the pair the minimizes $\mathrm{MDL}(\hat{h}, \hat{k})$ as in (4.2). We adopted a grid-search approach to minimize $\mathrm{MDL}(\hat{h}, \hat{k})$. That is, we compute $\mathrm{MDL}(\hat{h}, \hat{k})$ for every possible combinations of (h, k) for which $1 \leq h \leq h_{\max}$ and $1 \leq k \leq k_{\max}$, and the pair that gives the smallest value of $\mathrm{MDL}(\hat{h}, \hat{k})$ is taken as our final estimates for (h, k). Throughout our numerical work we set $h_{\max} = k_{\max} = 3$.

For a given value of (h, k), the calculation of the maximum likelihood estimates of the θ_i is straightforward. We illustrate this idea using the same 12×12 lattice given in Fig. 4, with a neighborhood of size 1×3; i.e., $h = k = 1$ and there are eight θ_is. Recall that θ_i controls the probability of the pixel being black for the i-th component of the local map. In this case we know the true values $\theta_0 = \theta_1 = \theta_4 = \theta_6 = 0.95$ while $\theta_2 = \theta_3 = \theta_5 = \theta_7 = 0.05$. Using (3.2), for the above 12×12 lattice we have $\hat{\theta}_1 = \hat{\theta}_4 = 17/18 = 0.944$, $\hat{\theta}_2 = \hat{\theta}_7 = 0$, $\hat{\theta}_3 = 2/16 = 0.125$, $\hat{\theta}_5 = 1/9 = 0.111$ and $\hat{\theta}_6 = 1$. As mentioned previously, if $\hat{\theta}_i$ is 0 or 1, then it will be replaced with 0.0001 or 0.9999, respectively. Once these $\hat{\theta}_i$s are obtained, calculation of $\mathrm{MDL}(\hat{h}, \hat{k})$ is straightforward.

B. Simulation Study

Numerical experiments were conducted to evaluate the practical performance of the neighborhood size selection criterion $\mathrm{MDL}(\hat{h}, \hat{k})$. A minimal check for the method proposed here is to verify whether it can recover the unknown rule of a SCA from a pattern generated using the respective SCA. To this purpose, we generate first an observed binary image from a known SCA rule and then we apply the proposed method to estimate h, k and the θ_is. The generated images are of size $M \times N = 512 \times 512$. Altogether we tested the method on eight different rules. The neighborhood sizes of these eight rules are listed in Table I. For Example, Rules 1 and 2, the corresponding θ_i values are listed in Table II, while for Example Rules 3 to 8, the θ_i values can be obtained from the authors. The corresponding generated observed images are given in the left columns of Figs. 5–8. For each of the eight different rules, the procedure was repeated

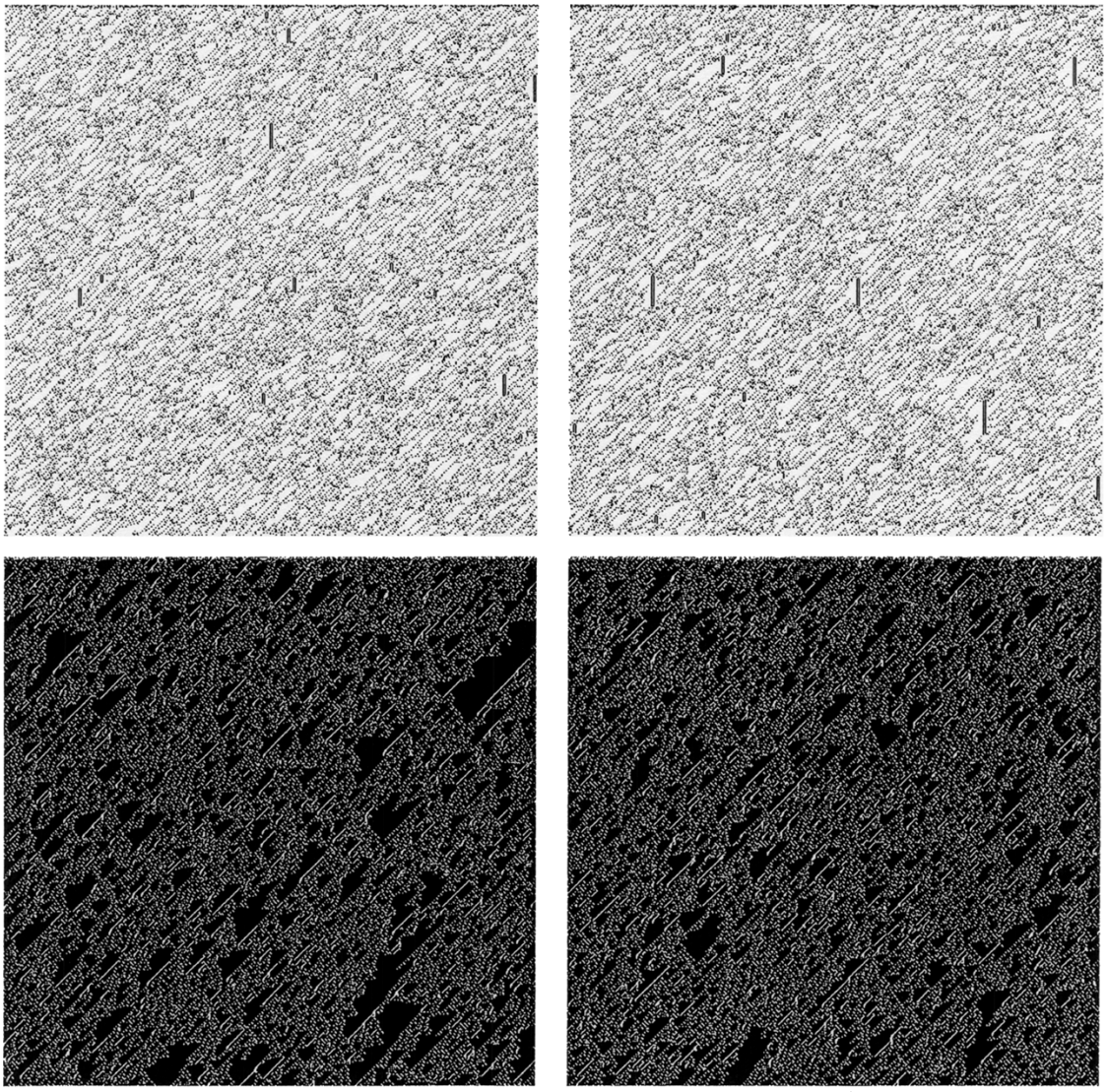


Fig. 8. Similar to Fig. 5, on the left are images generated from Example Rules 7 and 8; on the right are typical images simulated from rules that are estimated from the corresponding images in the left column.

500 times so that the effectiveness of the proposed estimation method could be assessed. In each of the 500 replicates, the local map is the same but we start the generation with different initial states. These initial states were generated at random from a distribution that assigns with equal probability the value 0 (white) or 1 (black) to each cell situated in the top h rows of the image.

Also listed in Table I are the number of times that both h and k were simultaneously and correctly estimated. For Example, Rules 1 and 2, listed in Table II are the averages of all the estimated θ_is corresponding to those repetitions for which the neighborhood size was correctly identified. For visual comparison, images simulated from the estimated rules are displayed in the right columns of Figs. 5–8. From these images and Tables I and II, the effectiveness of the proposed method clearly results.

C. Real Data

We have applied our method for various general patterns encountered in nature with some degree of success. It is also clear, from those trials performed, that certain patterns are more suitable than others. Fig. 9 displays two different patterns for which similarities between the original image and the reproduced one are quite different. In this respect, the method based on CA is no different than other pattern synthesis methods (e.g., [6] and [20]), which are suitable for some textures, but certainly not for all.

VI. Conclusion and Future Work

We present a nonparametric likelihood-based approach to reproduce patterns using stochastic cellular automata. The method works very well when we need to recover the rule from an observed realization of a SCA with a medium-sized rule. Such patterns are usually difficult to restore using other pattern generation methods. The method also works well with many regular patterns encountered in nature.

Nevertheless, in order to apply the same approach to a wider variety of binary patterns, we need to expand our method and to develop computational implementations so that neighborhoods

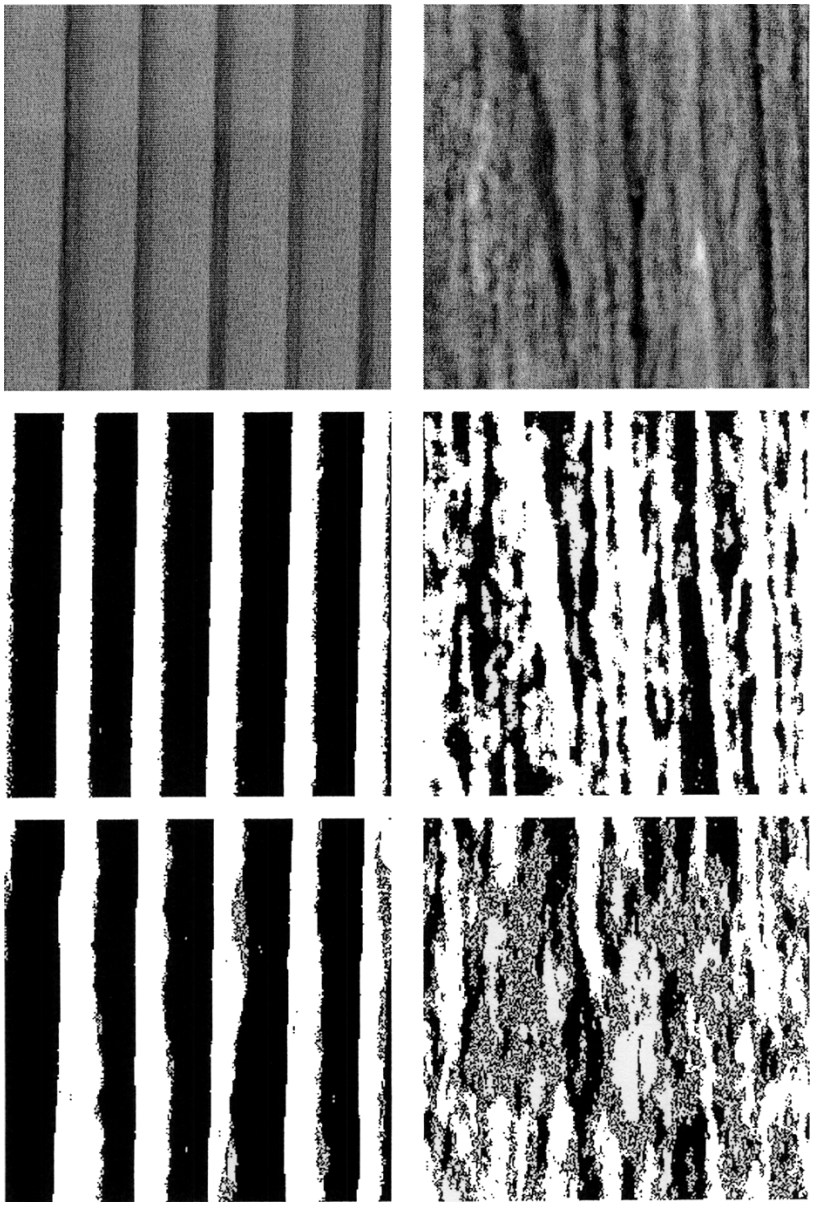


Fig. 9. Results from real patterns. Top row: Two real grayscale images. Middle row: Binary images obtained from thresholding the grayscale images in the top row (pixels with greyvalues higher than the threshold were set to black, otherwise they were set to white). Bottom row: Simulated images obtained from applying the presented method to the binary images in the middle row. Both patterns have the same estimated neighborhood size with $\hat{h} = 3$ and $\hat{k} = 2$.

of larger size can be handled. Another natural extension is to incorporate grayscale patterns. However, such an extension cannot be made in a straightforward manner since the dimensionality of the space of rules increases too fast to be manageable even for medium-sized neighborhoods like the ones considered here. We are currently investigating both of these issues.

Last, we can only hope that our approach adds another tool to an already rich gallery of methods which makes the choice of the right procedure a matter of art as much as it is a matter of science.

ACKNOWLEDGMENT

The authors would like to thank the referees and the Associate Editor for their most constructive comments.

REFERENCES

[1] A. W. Burks, Ed., *Essays on Cellular Automata*. Urbana, IL: Univ. Illinois Press, 1970.

[2] P. Garcia-Sevilla and M. Petrou, "Classification of binary textures using the 1-D Boolean model," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1457–1462, Oct. 1999.

[3] M. H. Hansen and B. Yu, "Model selection and the principle of minimum description length," *J. Amer. Statist. Assoc.*, vol. 96, pp. 746–774, 2001.

[4] T. E. Ingerson and R. L. Buvel, "Structure in asynchronous cellular automata," *Phys. D*, vol. 10, pp. 59–68, 1984.

[5] T. C. M. Lee, "An introduction to coding theory and the two-part minimum description length principle," *Int. Statist. Rev.*, vol. 69, pp. 169–183, 2001.

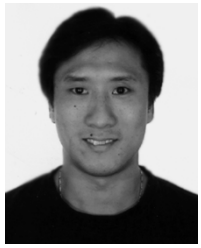
[6] T. C. M. Lee and M. Berman, "Nonparametric estimation and simulation of two-dimensional Gaussian image textures," *Graph. Mod. Image Process.*, vol. 59, pp. 434–445, 1997.

[7] Y. C. Lee, S. Qian, R. D. Jones, C. W. Barnes, G. W. Flake, M. K. O'Rourke, K. Lee, H. H. Chen, G. Z. Sun, Y. Q. Zhang, D. Chen, and C. L. Giles, "Adaptive stochastic cellular automata: theory," *Phys. D*, vol. 45, pp. 159–180, 1990.

[8] M. Madjarova, M. Kakuta, T. Obi, M. Yamaguchi, and N. Ohyama, "Two-dimensional cellular automata for pseudo-random pattern generators and for highly secure stream ciphers," *Opt. Rev.*, vol. 5, pp. 143–151, 1998.
[9] K. Preston and M. J. B. Duff, *Modern Cellular Automata: Theory and Applications*. New York: Plenum, 1984.
[10] F. C. Richards, T. P. Meyer, and N. H. Packard, "Extracting cellular automaton rules directly from experimental-data," *Phys. D*, vol. 45, pp. 189–202, 1990.
[11] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific, 1989.
[12] W. Turin, "Fitting probabilistic automata via the EM algorithm," *Commun. Statist. Stochastic Models*, vol. 12, pp. 405–424, 1996.
[13] S. M. Ulam, "Random processes and transformations," in *Proc. Int. Congr. Math.*, vol. 2. Providence, RI, 1952, pp. 264–275.
[14] J. von Neumann, *Theory of Self-Reproducing Automata*, A. Burks, Ed. Urbana, IL: Univ. Illinois Press, 1966.
[15] S. Wolfram, "Universality and complexity in cellular automata," *Phys. D*, vol. 10, pp. 1–35, 1984.
[16] ——, "Random sequence generation by cellular automata," *Adv. Appl. Math.*, vol. 7, pp. 123–169, 1986.
[17] S. Wolfram, *A New Kind of Science*. Champaign, IL: Wolfram Media, 2002.
[18] J. Xie, D. Zhang, and W. Xu, "Spatially adaptive wavelet denoising using the minimum description length principle," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 179–187, Feb. 2004.
[19] Y. X. Yang and S. A. Billings, "Neighborhood detection and rule selection from cellular automata patterns," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 30, no. 3, pp. 840–847, Jun. 2000.
[20] S. C. Zhu, Y. N. Wu, and D. B. Mumford, "Minimax entropy principle and its application to texture modeling," *Neural Comput.*, vol. 9, pp. 1627–1660, 1998.

Radu V. Craiu received the B.Sc. degree in mathematics and the M.Sc. degree in statistics in 1995 and 1996, both from University of Bucharest, Bucharest, Romania, and the Ph.D. degree in statistics from the University of Chicago, Chicago, IL, in 2001.

He is currently an Assistant Professor with the Department of Statistics, University of Toronto, Toronto, ON, Canada. His research interests include Markov chain Monte Carlo methods, competing risks models, statistical genetics, model selection, and image processing.



Thomas C. M. Lee received the B.Appl.Sci. degree in mathematics with University Medal from the University of Technology, Sydney, Australia, in 1992 and 1993, respectively, and the Ph.D. degree jointly from Macquarie University and CSIRO Mathematical and Information Sciences, Sydney, in 1997.

Currently, he is an Associate Professor at the Department of Statistics, Colorado State University, Fort Collins. His research interests include computational statistics, wavelet analysis, and digital signal and image processing.