Generating textile designs using cellular automata

S Rajasekaran* & R Amalraj
Department of Civil Engineering, PSG College of Technology, Coimbatore 641 004, India
Received 19 April 2001; revised received and accepted 14 September 2001

Cellular automata (CA) are discrete dynamical systems of simple construction but complex and varied behaviour. Algebraic techniques are used to give an extensive analysis of the global properties of a class of finite cellular automata. The rule numbers in one-dimensional automata ranging from 0 to 255 have been tested and found to generate mostly geometric patterns. In two-dimensional automaton, the various rule numbers have been tested in 80x80 matrix with a grid size of five pixels and the same size extended to 110x110 with a grid size of three pixels each that resulted in chaotic, stable and high life. The behaviour of the neighbourhood cells has been analyzed by their categories, grouped together and represented graphically, resulting in uniform and non-uniform patterns, which may well be utilized for textile designs. The algorithm is designed with an option to generate the rule number itself randomly and hence the design generated is highly unpredictable and it is observed that different patterns are generated to each iteration of the algorithm.

Keywords: Cellular automaton, Textile design

1 Introduction
The evolution of cellular automata (CA) can be thought of processing the data with their initial conditions, which is related to many models such as partial differential equations. In a cellular automaton, the sites would have just a few discrete values, where the evolution leads to effective randomization of individual site values but maintains smooth macroscopic average behaviour. Standard cellular automata have definite rules for updating the values of sites. Often, however, their microscopic behaviour can seem to be apparently random and can mimic the irregularity. A cellular automaton consists of a line of cells or sites, each with value 0 or 1. These values are updated in a sequence of discrete time steps according to a definite fixed rule. Denoting the value of a site at position \( i \) by \( a_i \), a simple rule gives its new value as:

\[
d_i = \phi(a_{i-1}, a_i, a_{i+1})
\]  

where \( \phi \) is a Boolean function which specifies the rule and, in general, the sites in a cellular automaton may have any finite number \( k \) of possible values. The rules for updating these sites may depend on a value up to any finite distance \( r \) away. In addition, the sites may be arranged not on a line, but on a regular lattice in any number of dimensions. CA are the mathematical idealization of physical systems in which space and time are discrete, and the physical quantities take on a finite set of discrete values. CA evolves in discrete time steps, with the value of the variables at one site being affected by the values of variables at sites in its neighbourhood on the previous time step. The neighbourhood of a site is typically taken to be the site itself and all immediately adjacent cells. The variables at each site are updated simultaneously, i.e. synchronously, based on the values of the variables in their neighbourhood at the preceding time step and according to a definite set of local rules. Physical systems containing many discrete elements with local interactions are often modelled as cellular automata. The development of structure and patterns in the growth of organisms often appears to be governed by very simple local rules and is therefore potentially well described by a cellular automaton model. The discrete values at each site typically label types of living cells, approximated as growing on a regular spatial lattice.

2 One-dimensional Automata

Let \( a_i^{(n)} \) denotes the value of site \( i \) in a one-dimensional cellular automaton at time step \( t \). Each site value is specified as an integer in the range \( 0 \) through \( k-1 \). The site values evolve by iteration of the mapping:

\[
a_i^{(t+1)} = F[a_{i-1}^{(t)}, a_i^{(t)}, a_{i+1}^{(t)}]
\]  

*To whom all the correspondence should be addressed.
Phone: 572177, 572477; Fax: 0091-0422-573853;
E-mail: sekaran@hotmail.com
where $F$ is an arbitrary function which specifies the cellular automaton rule. The parameter $r$ determines the range of the rule. The region affected by a given site grows by at most $r$ sites in each direction at every time step. The propagating features generated in cellular automaton evolution may therefore travel at most $r$ sites per time step. After $t$ time steps, a region of at most $1+2rt$ sites may therefore be affected by a given initial site value. Thus, the alternative form of the above equation is:

$$a_{i}^{t(n)} = F \left[ \sum_{j=r}^{r} \alpha_{j} a_{i-j}^{n-1} \right] ... (3)$$

where $\alpha_{j}$ is the integer constant and the function $F$ takes a single integer argument. Consider exclusively one-dimensional cellular automata shown in Fig.1(a), with two possible values of the variables at each site (base 2) and in which the neighbourhood of a given site is simply the site itself and the sites immediately adjacent to it on the left and right called "elementary".

Let us specify a particular rule for an elementary cellular automaton. On the top row, all $2^3 = 8$ possible values of three variables in the neighbourhood are given, and below each one is given the value achieved by the central site on the next time step according to a particular local rule as shown in Fig.1(a), with each site taking value 0 or 1. The eight possible states of three adjacent sites are given on the upper line. The lower line then specifies a rule for the time evolution of cellular automata by giving the value to be taken by the central site of the three on the next time step. The time evolution of the complete cellular automata is obtained by simultaneous applications of these rules at each site for each time step. The rule given is modulo-two rule: the value of a site at a particular time step is simply the sum modulo-two of the values of its two neighbouring sites on the previous time step. The Boolean equivalent of this rule is therefore schematically stated as:

$$S_{n+1}(m) = S_{n}(m-1) \oplus S_{n}(m+1) \quad ... (4)$$

$$S_{n} = S^{-} \oplus S^{+} \quad ... (5)$$

where $\oplus$ denotes addition modulo-two. The Boolean representation of cellular automaton rules reveals that some rules are 'peripheral' in the sense that the value of a particular site depends on the values of its two neighbours at the previous time step, but not on its own previous value.

2.1 Application of One-dimensional Automata

The output generated in one-dimensional automata for rule number 90, also called modulo-two rule, is shown in Fig. 2. The rule number itself could be generated randomly, hence the pattern created will be different whenever the program is executed. The output generated for rule number 45 is shown in Fig. 3.

3 Two-dimensional Automata

There are several possible lattices and neighbourhood structures for two-dimensional cellular automata. Four cells (including its own) in the neighbourhood, known as Von Neumann neighbourhood, is shown in Fig. 4(a). Also eight cells (9 including its own), known as Moore neighbourhood, and 6 cells in a hexagonal pattern.
known as uniform neighbourhood, are shown in Fig. 4(b) and Fig. 4(c) respectively. The possible rules for various neighbourhoods are given by:

- Von Neumann (5 cells neighbourhood) = $2^{10} - 2^6 + 1$
- Moore (9 cells neighbourhood) = $2^{14} - 2^8 + 1$
- Uniform (6 cells neighbourhood) = $2^{18} - 2^{10} + 1$

A five-neighbour square cellular automaton then evolves with respect to one dimension, so known as Von Neumann’s neighbourhood as:

$$a_{i,j}^{(t+1)} = \phi[a_{i,j}^{(t)}, a_{i-1,j}^{(t)}, a_{i+1,j}^{(t)}, a_{i,j+1}^{(t)}, a_{i,j-1}^{(t)}] \quad \ldots (6)$$

In case of a special class of totalistic rules, the value of a site depends only on the sum of the values in the neighbourhood. These rules are conveniently specified by code:

$$c = \sum_{n} f(n)k^n \quad \ldots (7)$$

The two-dimensional window is defined under two topologies, namely island and torus. In the island topology, the window is surrounded by dead cells, with cell growth and patterns end at the edges of the window, whereas for the torus, the bottom row of cells is assumed to be neighbouring the top row and the left column is treated as if it was adjacent to the right side of the window which allows the patterns to wrap around the screen, top to bottom and left to right. The most common and popular cellular automata is elementary computerized imitation of a colony of living cells. The system is built on a two-dimensional grid; cells live, die and are born according to a simple set of rules:

**The Rule of Birth:** If exactly three live cells are adjacent to an empty cell, the empty cell comes to life.

**The Rule of Overpopulation:** When a living cell has four or more live neighbours, it dies from overcrowding.

**The Rule of Underpopulation:** Any living cell with one or no neighbouring cell dies.
3.1 Application of Two-dimensional Automata

The output generated by two-dimensional automata with 30 iterations and random values ranging for 1 to 9 for the neighbouring cells, which could be utilized as textile designs, are shown in Fig. 5. The number of iterations can be up to the maximum number of rows and columns defined in the matrix. When the input file is represented with many cells turned on, the design generated by the application of two-dimensional automata will be chaotic.
3.2 Behaviour of Neighbouring Cells

Life represents a seminal example of artificial life. Embodied in three absolute rules is a pocket universe, one that is completely under a person's control. The effect is almost hypnotic, watching images formed and moving in a universe defined inside a machine. Starting patterns face one of two fates in life—complete extinction or eternal repetition. The blinker will forever repeat two states, but glide causes it to copy itself along a diagonal line; given an infinite grid of cells, the glider will travel forever or until it encounters other live cells that disrupt its form.

The life rules could be restated as: (i) any live cell with two or three living neighbours comes alive, and (ii) any dead cell with anything other than three living neighbours will die.

The above rule can be stated as S23 / B3. Life has four parameters, viz. high and low values for birth and survival. Changing those parameters, say allowing a cell to survive with three or four neighbours, creates a new automaton with new dynamics. Life and brain belong to a class known as two-dimensional semi-totalistic automata. Thus, the qualitative classes of cell behaviour can be identified as (i) disappear, (ii) periodic, (iii) chaotic, and (iv) stable. If the rule S 5678 / B 35678 is applied, then the pattern is called "High Life". For example, life requires two or three neighbours for birth, three neighbours minimum (and maximum) for survival, and has only one echo state.

Brain requires exactly two neighbours for birth and has one refractory state; the minimum and maximum survival values can be set to nine (outside the possible neighbour count of one to eight) for brain to indicate that all live cells enter echo states.

4 Cellular Automata for Woven Fabric Design

At present, most Jacquard cloth manufacturers use modern ordinary harness machines providing complete versatility in respect of the type of structure produced and built in a fine pitch to give large figuring capacity. It would seem that the preparation of design for Jacquard machines would need to revert to the laborious system of painting in the lifting sequence of each end in the repeat in full. Instead, if a designer's sketch can be currently developed with a fully worked out design with the aid of a computer, which when linked with a card-cutting machine makes the whole process automatic for the development of various designs. It is seen from the present research that the cellular automata will be very useful in creating infinite variety of designs from a given design. The entire representation is made in the grid system of five pixels, three pixels, and there are several ways to make it bigger or smaller. As the pixel size is reduced, the designs created will be more attractive. Once the designs are obtained from the printer, there are various methods to transfer the outlines of a design ready for painting, depending on the fabric used. If the fabric is of a light colour and fairly sheer, place the design underneath the fabric, trace the design onto the fabric with a soft (4B) pencil or other outline marker, known as 'see through method'. Another method called 'window method' is used for slightly darker or denser fabrics, wherein the design is brought to a window and then the fabric is fixed smoothly over it. The strong light from behind should enable us to see the design clearly. There are other methods, namely 'spoon rubbing', 'Dressmaker's carbon paper' and 'cutting templates', to work further on the designs generated by the program.

5 Conclusion

The simple calculations in the concepts of neighbourhood in cellular automata lead to the generation of many patterns in one-dimensional automata based on the rule numbers ranging from 1 to 256. In the case of two-dimensional automata, the patterns are classified as chaotic, high life and stable with respect to the existence of the neighbouring cells and result in regular designs over the iterations, which are actually utilized as textile fabric designs.

Acknowledgement

The authors are thankful to the management of PSG College of Technology and to Dr S. Vijayarangan, Principal, PSG College of Technology, for providing the necessary facilities to carry out the research work reported in this paper.

References

10 Gardner M, Mathematical games, Scient Am, 226 (1972) 104.