Sensors & Transducers, Vol. 169, Issue 4, April 2014, pp. 223-227

Sensors & Transducers

http://www.sensorsportal.com

Application of One-dimensional Cellular Automata in Designing Fractal Pattern

Fan Chang Xing
Department of Computer Center, Shao Xing University, Shao Xing, Zhejiang, 312000, China
Tel.: 13345859960
E-mail: fcxjszj@usx.edu.cn

Received: 9 April 2014 Accepted: 28 April 2014 Published: 30 April 2014

Abstract: This article focuses on elaborating the principle of cellular automata and putting forward a new method to take advantage of the diversity and randomness of cellular automata to design printing design. Through this method we can provide a new rational creation tools for the design of the printing design, and finally we use JAVA programming to generate the printing design in computer.

Keywords: Cellular automata, Fractal pattern, Automatic design, One-dimensional, Cell, Rule.

1. Introduction

Industrial pattern design is a very long process, even using the computer, just use it as an advanced and modern brush, design conception is still limited in the field of "visual", depends entirely on his "subjective" to carry on the design. The powerful digital information of computer has not really get into graphic design process, the traditional mode of production has obvious shortcomings such as narrow design ideas, less flowers, long cycle, production of hard faults and so on, is no longer adapt to the pace of modernization and market demand. In this article, according to the principle of cellular automata machine, we put forward a new method of fractal pattern automatically design based on cellular automata. When user input some parameters, this system can generate the corresponding printing design and truly achieved the automatic design function of computer.

2. Brief Introductions of Cellular Automata

Ulm (Stanislaw M. Ulam) and von Neumann (John von Neumann) in order to study the possibility of the robot's self-replication, in the last century 50's, they put forward a Discrete Dynamical Systems called cellular automata (CA) [1-2]. The basic idea of Cellular Automata model: many complex structure and process in nature just be caused by the simple interaction of a large number of basic units in the final analysis [3]. Therefore, we can use a variety of cellular automata to simulate the evolution process of any complicated things. In this article, we will discuss the use of one-dimensional cellular automata generate printing design.

2.1. Characteristics of Cellular Automata

Parallel computing: Each cell should be able to synchronize operation by parallel processors to operate [4].

Local: The state of cells changes only by the close surrounding cells, the change of all cells need to set the cell value and change the operation rule.

Homogeneous: Each cell is guided by the same set of rules for the operation and influences each other.

2.2. The Basic Elements of Cellular Automata

The composition of CA elements include: grid, grid state, adjacent to state space, the evolution rules and so on [5-6].

The grid (cells): CA is constituted by a group of the grid (or cells). Theoretically these grids can be any geometric shape, and can even stereo unit, but at present the majority of CA studies are based on irregular grid, its spatial structure and grid-type data structure of the same pattern, and as a one-dimensional or two-dimensional matrix.

Grid state (states): Each grid is displayed by a limited set of state, and the range of these states can be binary, such as: live, dead; empty, has been occupied. It can also be a diverse collection of classes, such as: building sites, vacant land, commercial land, residential land and other land-use types. State can also be a real number or a range of values of the ranking scale. At any one time, each grid will be rendered only a specific value in the state of this group.

Adjacent area (neighborhood): the state of each grid in CA will change with the state of grid in its neighboring districts. So design of a CA needs to define the size of its influence mutually adjacent areas. To the grid-type structure of the data, adjacent areas can be centered grid nearest surrounding grid or within a certain distance of all grids.

Evolution rules: The state of each grid in the next time is determined by the total composition of its current patterns and the effects of its surrounding region grid. Evolution of Patterns of the next point is determined by a definite rule. Under the above-described space structure, the cycle of the evolution of the CA is in a discrete sequence of the time (..., t-1, t, t +1 ...), all the grid according to the evolution rule synchronous update.

3. The Theory of One-Dimensional Cellular Automata

The state set of most simple one-dimensional cellular automata has two elements {0, 1}. The neighbor is a region with one radius, that is, left and right two squares of one grid are its neighbors. Because the state set only has two elements {0, 1}, which means that each grid only have black or white color. any one grid with its two neighbors can combination 8 states, each respective of these eight kinds of combined state can determine the next cell to be black or white, according to this way, the number of arrangement can reach to 256. Therefore in the types of cellular automata considered by Wolfram, the number of rules on state of cell changing is 256, Wolfram laid out number for each rule. For example, Fig. 1 is a virtual representation of rule 110 [7-8].

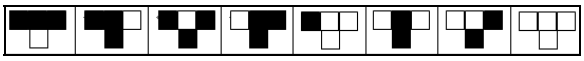

Fig. 1. Rule 110.

The top row shows eight color combinations of the cell and its left and right neighbors, the bottom row shows the next color of the middle cell. Text described: When the three adjacent cells on the line above to a cell are all black, all white, or the left side of these three adjacent cells is black, this cell is white, otherwise black.

Although a one-dimensional automatic cell machine is one-dimensional, but if we use a group of consecutive rows from top to bottom to display its evolution, we can get a pattern composed of some black and white grid. Fig. 2 shows the CA's first five steps of evolution with rule 110 (including the initial state). You can see that the color of each cell is decided by its own color and the color of the nearest neighbors in the previous line according to the rule 110, which means that the value of cell at time "t + 1" will only depend on the value of the cell and its left and right neighbor at time "t" [9-12]. Meanwhile, the value of all the cells in the same line is updated at the same time during the every step of evolution. Now, we design a CA with the edge of the cells as black and all other cells as white in the first line, when we use rule 110, this CA will evolution follow with a series of steps shown in Fig. 2.

Fig. 2. Sequence of steps of rule 110.

4. Fractal Graphics of One-Dimensional Cellular Automata

Aside from the cell's intrinsic time scale, which is of no concern in cellular automata, we will introduce an external clocking mechanism which resets the input u_i of each cell "C_i" at the end of each clock cycle by feeding back the steady state (i.e. attractor) output $y_i \in \{-1, 1\}$ as an updated input $u_i \in \{-1, 1\}$ for the next iteration. The resulting system is called a one-Dimensional Cellular Automata with a periodic boundary condition. Each cell "i" in this system has a state variable $x_i(t)$, an output variable $y_i(t)$ and three constant binary inputs u_{i-1}, ui, and u_{i+1},the state is shown in Fig. 3.

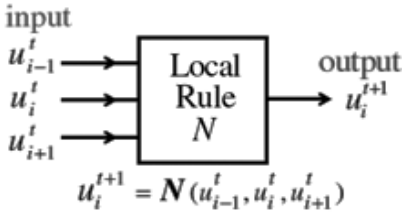

Fig. 3. The state of one cell.

Now we assume that all the squares are located in a straight line, and the length of the straight line is the width of the area of our animation, for example, if the length of this line is 400, which means that there are 400 squares on this line. We fill black in the grid on state "1" and fill white in grid on state "0", then an intermittent horizontal line is a distribution for the current state of all cells. These grids can form different lines with changing over time, and we put these lines together to form a grid area. Wherein the vertical axis represents the passage of time (downward direction is positive), the horizontal axis represents the state of the cellular automaton in the corresponding moment, through this way we can obtain an image.

When we draw graphics, we should draw the pixels of the first line according to a certain rule, and the pixels of the later line can get a value according to a certain operation regulations based on previous behavior, which can determine each pixel to be drawn or not, thereby forming a pattern. Now we set the initial state of a simple cell, the number of warp yarns is 10 and the number of weft yarn is 10 also. Black circle represents the warp yarns above the weft yarns, the white circle indicates the warp yarns under the weft yarns, the warp and weft yarns interwoven conditions at the initial state (t = 1) is shown in Fig. 4:

Fig. 4. Initial chart.

We derive 'the first step iterative map' from the begin of 'initial chart': the first to eighth point is white in 'initial chart', and these points retain the original white accordance with the provisions of iteration in 'the first step iterative map'; only the right point of the ninth point in 'initial chart' is black, so the ninth point will be changed into black according to the rule in 'the first step iterative map'; the left point of the tenth point is white and the tenth point itself is black in 'the initial chart', so the tenth point maintain the original black in 'the first step iterative map', now we have checked every point in 'the initial chart' and remember the new color of all the points in 'the first step iterative map', then we can draw 'the first step iterative map' according to these derivation in Fig. 5.

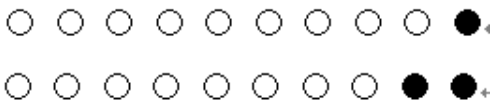

Fig. 5. The first step of iterative.

Followed this way, we can obtained an irregular organization chart with the loops of 10 × 10 warp yarn and weft yarn , the code data is shown in Fig. 6. Fig. 7 is the fabric organization chart and a flat-screen effect chart based on code data in Fig. 6.

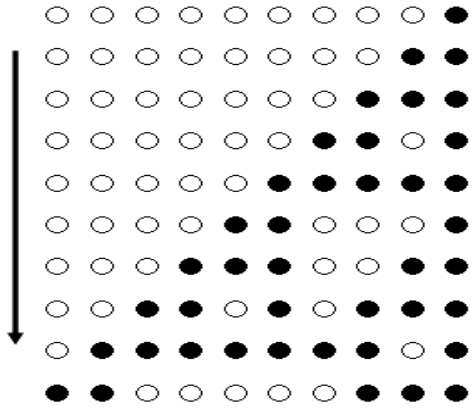

Fig. 6. Code data chart.

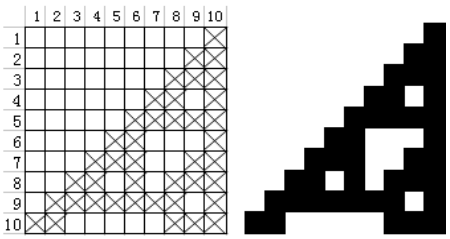

Fig. 7. The fabric organization chart and a flat-screen effect chart.

Creating the fractal requires little computational power, even simple graphing calculators can easily make this image. The fractal is created with pixel to pixel by using random numbers and the fractal will be slightly different each time due to this. If you were to run the program repeatedly and allow each to use an infinite amount of time, the results would be always identical. No one has an infinite amount of time, but the differences in the finite versions are very small.

To generate this fractal, a few steps are involved. First, initial X and Y values should be chosen, either by the program or the user. The values used have little effect on the fractal. Regardless of what's chosen, the same triangle will be created. Next, the program must create a random number, between 0 and 1. Then, three possible routes can be taken.

5. The Implementation of System

A Boolean function is usually described in computer science or informatics by a truth table where each binary variable is represented symbolically by either a "0" or a "1". In this paper, it is absolutely essential that we use "−1" and "1" instead of "0" and "1" because these variables, except in a few strictly Boolean settings, must be interpreted as real numbers in all subsequent mathematical analysis and calculations, such as solving differential equations, which are all based on the real number system [13-15]. Hence, the truth table for a Boolean function of three binary variables u_{i-1}, u_i, and u_{i+1} will be depicted in Fig. 8.

The only exception to this assumption is in the output y_i of cell "C_i" where we may revert back to "0" and "1" whenever it does not enter into any arithmetic or algebraic calculations. For example, it is more convenient to decode the output $y_i = (\gamma_7, \gamma_6, \gamma_5, \gamma_4, \gamma_3, \gamma_2, \gamma_1, \gamma_0)$ in decimal system by recasting it into its equivalent binary form $y_i = (\beta_7, \beta_6, \beta_5, \beta_4, \beta_3, \beta_2, \beta_1, \beta_0)$ where β_j is either a "0"or "1" so that the corresponding decimal number is simply the integer $N = \beta_7 \cdot 2^7 + \beta_6 \cdot 2^6 + \beta_5 \cdot 2^5 + \beta_4 \cdot 2^4 + \beta_3 \cdot 2^3 + \beta_2 \cdot 2^2 + \beta_1 \cdot 2^1 + \beta_0 \cdot 2^0$,as shown in Fig. 8. Since there are 2^8= 256 distinct combinations of this eight-bit word, there are exactly 256 distinct Boolean functions, each one identified uniquely by an integer N, where N = 0, 1, 2, . . . , 255. It is important to observe that the output y_i specifies either a Boolean rule (when coded in "−1" and "1") or its identification number (when coded in "0" or "1").

Truth Table

$y_i = N(u_{i-1}, u_i, u_{i+1})$

	u_{i-1}	u_i	u_{i+1}	y_i
0	-1	-1	-1	y_0
1	-1	-1	1	y_1
2	-1	1	-1	y_2
3	-1	1	1	y_3
4	1	-1	-1	y_4
5	1	-1	1	y_5
6	1	1	-1	y_6
7	1	1	1	y_7

$y_i = (\gamma_7, \gamma_6, \gamma_5, \gamma_4, \gamma_3, \gamma_2, \gamma_1, \gamma_0)$

Binary code: $\gamma_i \in \{-1, 1\}$

$(\beta_7, \beta_6, \beta_5, \beta_4, \beta_3, \beta_2, \beta_1, \beta_0)$

Decimal code: $B_j \in \{-1, 1\}$

$N = \beta_7 \cdot 2^7 + \beta_6 \cdot 2^6 + \beta_5 \cdot 2^5 + \beta_4 \cdot 2^4 + \beta_3 \cdot 2^3 + \beta_2 \cdot 2^2 + \beta_1 \cdot 2^1 + \beta_0 \cdot 2^0$

Fig. 8. Boolean function for y_i.

Since the three binary variables u_{i-1}, u_i, and u_{i+1} in a one-dimensional Cellular Automata are coded in terms of real numbers "−1" and "1", we can identify each input (u_{i-1}, u_i, u_{i+1}) as a vertex of a cube (of length 2 on each side) centered at the origin. It is also extremely convenient to refer to each of these eight vertices by an integer 0, 1, 2, ... , 7, by reverting back to its corresponding three-bit binary word. For example, the binary word associated with the vertex located at (−1, 1, 1) is 011, which decodes into the integer 3. In other words, we can identify uniquely each vertex of the Boolean cube by an integer n, where n ranges from 0 to 7.

We will formally identify each local rule by a dynamical system defined as follows:

State Equation:

$$\begin{aligned} \dot{x}_i &= f(x_i; u_{i-1}, u_i, u_{i+1}) \\ x_i(0) &= 0 \end{aligned}, \tag{1}$$

Output Equation:

$$y_i = y(x_i) \triangleq \frac{1}{2}(|x_i + 1| - |x_i - 1|), \tag{2}$$

Such as, use this equation, we can define the rule 110's equation:

State Equation:

$$\dot{x}_i = -x_i + 2y_i + [-2 + |(u_{u-1} + 2u_i - 3u_{i+1} - 1)|], \tag{3}$$

According to this algorithm, we design program interface, the input dialog box for the parameters of cellular automata is shown in Fig. 9, in which state expresses the number of states each cell will have; radius expresses the number of neighbors each cell will have, code expresses which the rule does the cellular automata use; type expresses length coding or short coding.

When we set the radius of "1" and the rule of "110", the graphics produced by one-dimensional Cellular machine is shown in Fig. 10.

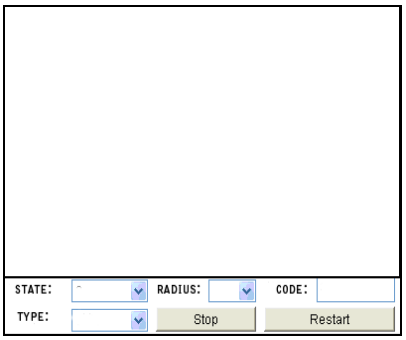

Fig. 9. The input dialog box.

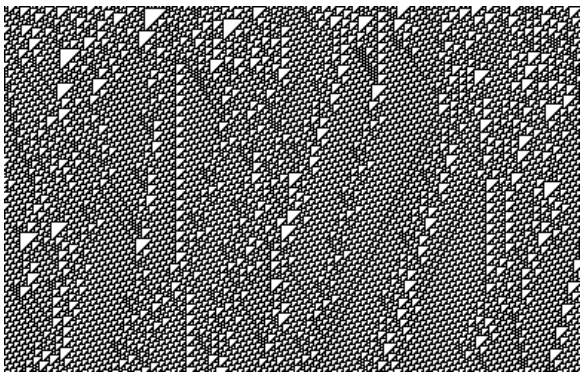

Fig. 10. Fractal graphics generated with the radius "1" and rule "110".

The device of this cellular automaton is shown in Fig. 11, which include the LCD screen, motherboard, keyboard, control panel, storage management, equipment and accessories. The accessory is a component of the CA program. For this device, if we give seed cells and rules, and then according to these rules, cells can automatic propagation, when we stop the device at an appropriate position, we can get wonderful fractal graphics.

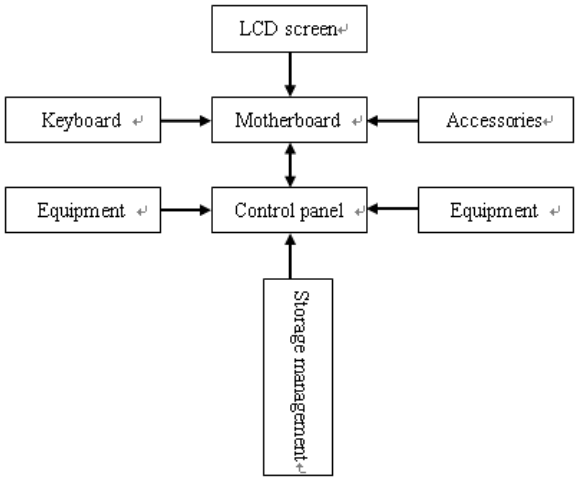

Fig. 11. System structure diagram.

6. Conclusions

Using the one-dimensional cellular automata theory to automatically generate irregular fabric organization with the conversion of the rules and the change of neighbor radius, we can obtain more patterns, and these patterns are suitable for, which provide a rich source for fractal pattern design.

Acknowledgment

This work was supported in part by a grant from 2013 scientific research fund of Zhejiang province of China education department (Y201327861), Shaoxing, Zhejiang province science and technology projects of China (2013B7006).

References

[1]. Wolfram S., Theory and Application of Cellular Automata, *World Scientific,* Singapore, 1986.

[2]. Kari J., Theory of Cellular Automata, *Theoretical Computer Science*, Vol. 334, No. 1, 2005, pp. 3-33.

[3]. Yongning Zhang, Yingjin Gan, Dongsheng Chen, The computer-aided design of irregular organization, *Journal of Textile Research*, Vo1. 25, No. 6, 2004, pp. 110-111.

[4]. Yang Xiaoli, The study on the characteristics of a chaotic Cellular Automata, *Northeast Normal University,* 2009.

[5]. Hu Yue, Sun Jiang Lin, Zhou Qing, Application of 2D Cellular Automata in Image Authentication, *Computer Engineering*, Vol. 36, No. 23, 2010, pp.110-112.

[6]. Han Zhuo-Bing, Ni Sheng-Qiao, Water wave animation synthesis based on cellular automata and smooth filtering, *Journal of Computer Application*, Vo1. 30, Supp. 1.2, 2010, pp. 199-201.

[7]. Paul Reiners, Cellular Automata and Music, *Developer Works,* China, 2004.

[8]. Gan Tao, The computationalism of the cellular automata and modern science, http://www.mostai.com/modules/article/view.article.php /28, 2007

[9]. Zhang Rui, The application of fractal geometry in computer hosiery machine, *Zhejiang University of Technology,* China, 2008.

[10]. Chen Chen, Chen Jianqiao, Cellular Automata Based Traffic Flow Simulation and Actuated Signal Control Strategy, *Journal of Highway and Transportation Research and Development*, Vol. 28, No. 6, 2011, pp. 122-127.

[11]. Shi Ningguo, Jie Yaping, Study on System of Forecast Urban Land Usage Based on Cellular Automata Model, *System Simulation Technology*, Vol. 6, No. 3, 2010, pp. 192-196.

[12]. Jiang Haiku, Zhen Chunking, Liu Sufi, Printing patterns design based on Mandelbrot set in fractal theory, *Journal of Textile Research*, Vol. 31, No. 12, 2010, pp. 139-142.

[13]. Kang Zhen-Human, Wang Hua-Yu, Design of Pattern Based on Two-Dimensional Cellular Automata, *Computer Technology and Development*, Vol. 22, No.1, 2012, pp. 111-113.

[14]. Hong Pan, Lou Ringlet, Chen Hue, Studies on dyeing and weaving pattern design based on mathematical pattern, *Modern Textile Technology*, No. 3, 2012, pp. 18-21.

[15]. Leon O. Chua, A nonlinear dynamics perspective of wolfram's new kind of science. Part 1: Threshold of Complexity, *International Journal of Bifurcation and Chaos*, Vol. 12, No. 12, 2002, pp. 2655–2766.