

A Technique for Describing and Transforming Images Based on the Evolution of Cellular Automata

Stepan Bilan

State University of Infrastructure and Technology, Kyrylivska, 9, Kyiv, 04071, Ukraine
Taras Shevchenko National University of Kyiv, Volodymyrska Street, 60, Kyiv, 01033, Ukraine

Abstract

The paper describes the forms of representation of the evolution of one-dimensional and two-dimensional cellular automata. The possibility of describing and presenting images using elementary cellular automata is shown. Two forms of representing the evolution of two-dimensional cellular automata are described, which have different lengths of time steps when constructing color codes in each cell. Evolutions of two-dimensional cellular automata are formed on the basis of Wolfram coding for four and five cells forming a neighborhood. The first option uses evolution for 24 time steps, which makes it possible to form color images with RGB color representation. In the second version, decimal color coding is used, which also makes it possible to form color images, while the cell codes are preserved and changed only when evolution changes them at the next time steps. The main theoretical principles of the formation of the evolution of two-dimensional cellular automata on one two-dimensional cellular field in the form of a color image are described. Principles of simple image preprocessing operations are presented using the example of image shift, and the possibility of using various transition rules for constructing an image of any complexity is shown. Representation of an image using a sequence of rules allows you to form an effective base of standards, which allows to easily recognize images, both binary and color. Analysis of neighboring cellular automata making up binary slices of a color image makes it possible to efficiently recognize them.

Keywords ¹

Image, cellular automata, transition rule, Wolfram coding, evolution, image shift, neighborhood of cells

1. Introduction

Full-flowing rivers are formed from small streams.

The current state of the art of image processing and recognition is characterized by a variety of methods and tools that differ in accuracy, speed, as well as models and paradigms [1 - 7]. Most image recognition methods are aimed at recognizing specific geometric shapes (image of text, fingerprint, face, license plate, barcode, moving objects, etc. images) [6]. Such methods implement a specific algorithm, which includes image preprocessing operations that implement image normalization, noise removal, and other operations. These methods and the means that implement them have fairly high rates of accuracy and speed. However, almost all existing image recognition methods cannot claim universality, since, due to their complexity, color images cannot be described by a simple set of informative features. If complex color images are described with a set of informative features, then there can be a large number of such features. In this case, the description of the image becomes more complicated. In addition, different encoding of raster images leads to their ambiguity when describing their characteristic features in different systems.

II International Scientific Symposium «Intelligent Solutions» IntSol-2021, September 28–30, 2021, Kyiv-Uzhhorod, Ukraine

bstepan@ukr.net

ORCID: 0000-0002-2978-5556



© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

An important aspect in the creation of technical vision systems is the effective description of images, its representation is read with a minimum set of simple features, with the help of which its complete restoration is carried out. In addition to the set of features, the image can be described by other methods. The most modern methods are methods based on representing an image as a sequence of shifts in different directions [6]. As a result of such shifts, sets of quantitative characteristics are formed (mainly in terms of the area parameter). The convenience of such methods lies in the homogeneity of the describing parameter. Other approaches are also possible, implemented on artificial neural networks and other paradigms.

Specialists pay much attention to the description and processing of images based on cellular automata (CA) [7, 8]. For this, various forms of coverage and forms of surroundings are used. One-dimensional (elementary) CA (ECA) [9, 10] and two-dimensional CA (TKA) [12 - 14] are used, and their initial states and evolution at a certain specified number of time steps are also taken into account. The flexibility and reconfigurability of the cellular automaton structure allows realizing the dynamics of changes in the characteristic features of any two-dimensional picture, which allows a detailed analysis of images to understand them. The three-dimensional construction of the CA has more functionality. The location of a set of TCA on one straight line allows one to obtain a three-dimensional cellular structure, which makes it possible to form a binary code for each cell, the dimension of which is determined by the number of TCA. The construction of such a two-dimensional code makes it possible to "see" any ECA in the sequence. The code of each cell of the two-dimensional array visually forms a color, which makes it possible to determine the state of each ECA in the evolutionary sequence. In this case, it is also necessary to take into account the structure of the code that encodes the color characteristics of each pixel. It is necessary to take into account the number of bits it takes to represent one of the three colors (blue, green, red).

This paper solves the problem of describing color images based on the analysis of the evolution of two-dimensional cellular automata, which allows efficient processing of images of any complexity. Cellular automata are used to describe bitmap images in RGB format, where each color is represented by one byte. In addition, cellular automata are used to simulate various dynamic processes over time. This is especially true for modeling the interaction of many objects (for example, cells or particles of matter) in time.

2. Evolution of Elementary CA

A cellular automata is defined as a lattice at the nodes of which finite automata (cells) are located, which have homogeneous connections with the nearest neighbors. If the lattice is one-dimensional (cells are located on one line), then the CA is called one-dimensional or elementary. ECA is fully described by Stephen Wolfram, who investigated their behavior in detail and made practical recommendations for their application [9 - 11]. ECA is a line of cells (finite automata) that are homogeneously interconnected. Each ECA cell at each time step of the iteration has one of two states (logical "1" or "0") and implements a local logical function of cell transitions to the next state. The arguments of the logical transition function are the states of the neighborhood cells. Boolean function defines a rule that can be represented by encoding rules. These transition rules are collectively referred to as Wolfram coding [9]. The ECA uses the states of three neighboring cells to execute the transition rule. In this case, on the basis of the states of the three analyzed cells, the state is changed by the middle cell. In this way, the states of all ECA cells change.

There are various options for the neighborhood for ECA, but the most popular are the classic ones, consisting of three cells (own, right and left cells). For such neighborhoods, Wolfram described 256 rules for the transition of a cell state at the next time step [9, 10]. The entire number of the described ECA state transition rules is limited to three cells and eight possible coding combinations.

Thus, for Wolfram encoding, eight sets of combinations are used, each number of which determines the number of the rule, as well as the binary code of transitions. For example, rule $111_{10}=01101111_2$. Each bit of the code of rule 111 is represented by a code that is formed by three bits of cells. An example of such coding for rule 111 is presented in Table 1.

Wolfram coding simplifies the process of implementing transitions and makes it possible to quite simply represent the evolution of ECA changes in time. Until now, practically all the rules have been

investigated and the class of their use for solving various problems in modeling has been determined. More and more works are devoted to the study of various neighborhoods in ECA. Various functions and their combinations are also investigated to simulate various dynamic processes.

How does ECA evolution is presentation?

All ECA states are formed in the form of lines of cells located from top to bottom as the value of time steps changes. The initial state is displayed at the zero step in the upper ECA. Examples of ECA evolutions for rules 55, 111, 222, as well as for identical initial states in Figure 1 are shown.

Table 1

Transition Rule 111 for ECA

The state of the cell $a_i(t)$ at the next time step	Neighborhood cells state at the previous time step $a_{i+1}(t), a_i(t), a_{i-1}(t)$
1	000
1	001
1	010
1	011
0	100
1	101
1	110
0	111

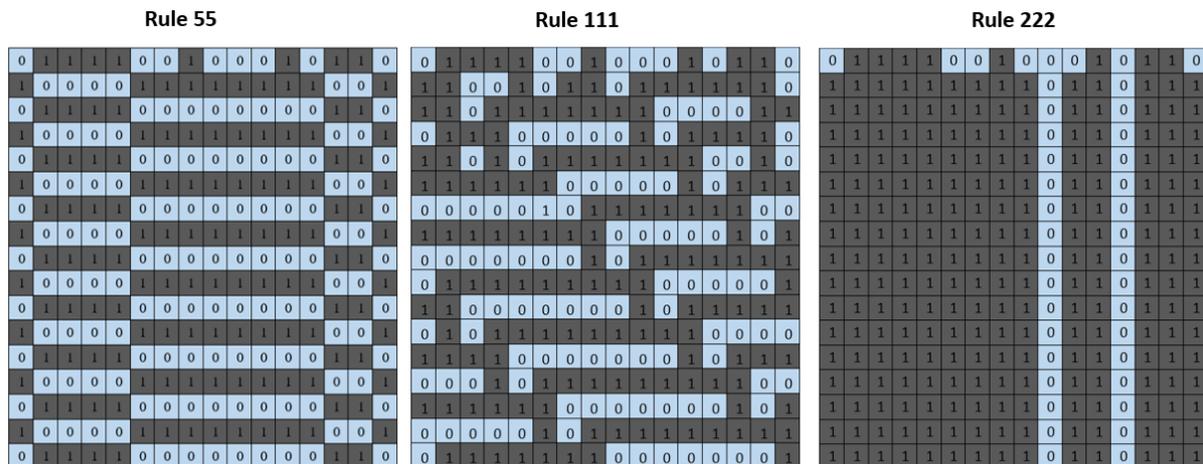


Figure 1: Evolutions of ECA behavior for rules 55, 111, 222 and for identical initial states of cells of cellular automata

As can be seen from Figure 1 evolutions are dramatically different for all the rules presented. ECA evolutions are represented by a two-dimensional array of cells, which we can consider as a binary image. By manipulating the initial states and rules, it is possible to form and describe any image. Thus, a two-dimensional binary image can be represented by the initial state of the ECA and a sequence of rules. An example of an image with the same initial conditions and a sequence of rules 55→111→222 in Figure 2 is shown. It is possible to effectively select suitable rules for modeling the required dynamic process, as well as for building a two-dimensional picture of a given structure.

ECA evolution shown in Figure 2 differs from the ECA evolutions shown in Figure 1. However, only three transition rules are used (sequential alternation of these rules every three time steps of the iteration). Figure 2 on the right shows the alternating rules for ECA opposite each iterative step of the CA. If you use more rules and their smaller localization in the ECA field, then you can successfully form and describe any closed lines and complex binary images. In this case, it is necessary to know

the initial structure of the image, which is a serious disadvantage for ECA. Unlike ECA, two-dimensional CA have more functionality in image processing.

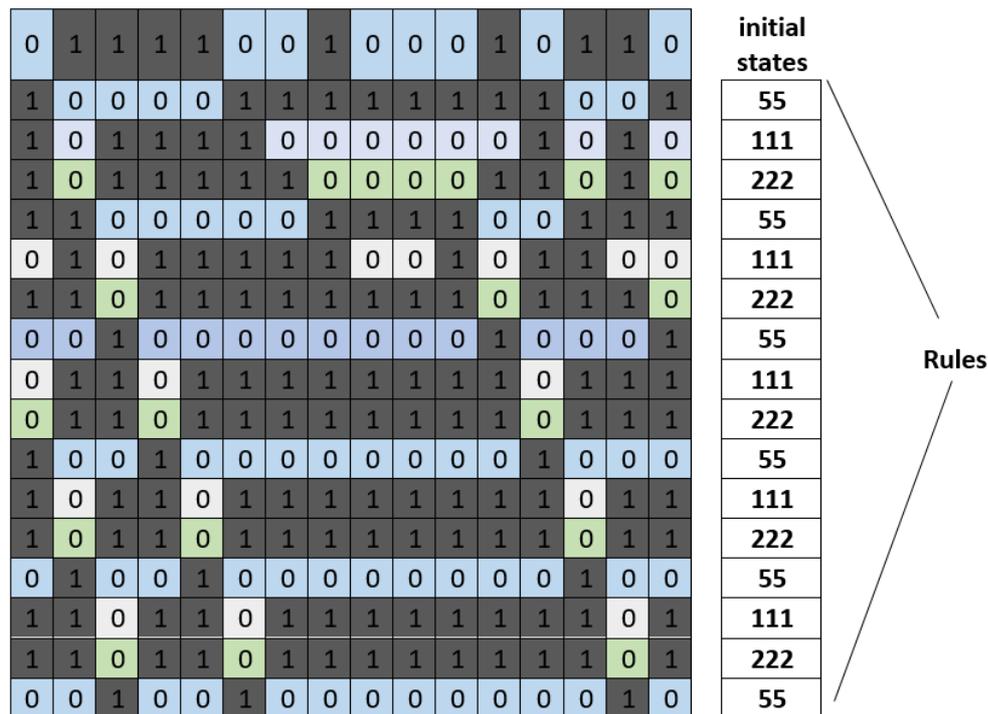


Figure 2: Evolution of an elementary cellular automaton with a sequential and repeating combination of rules 55→111→222 for the initial conditions shown in Fig. 1

3. New Forms of Evolution of Two-Dimensional Cellular Automata

The evolution of two-dimensional cellular automata may not be so visualized as elementary CA. Today, the evolution of TCA is most often represented as a sequence of two-dimensional pictures of TCA states, each of which reflects the state of TCA cells at a certain time step of evolution [15]. An example of such evolution in Figure 3 is shown.

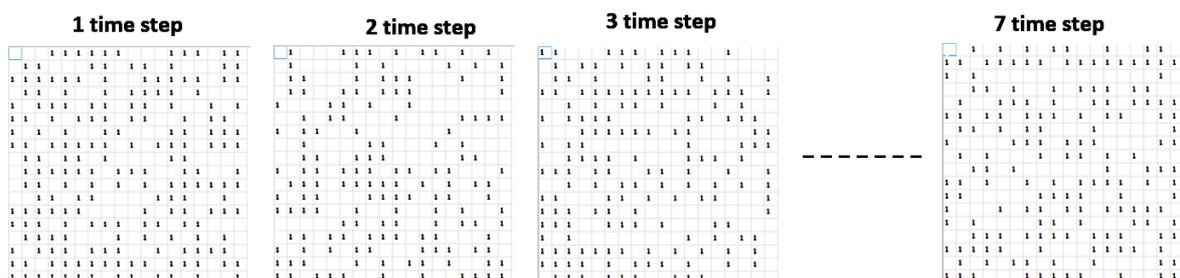


Figure 3: An example of TCA evolution in the form of a sequence of two-dimensional pictures

Such a mapping of the TCA evolution (Figure 3) is complex and does not clearly reflect the dynamics of state changes. It is very difficult for the user to trace changes in evolution. It is also necessary to use complex algorithms for comparing states for all iterations of states of cellular automata. If all the obtained TCAs at each time step are arranged sequentially one after the other and for each cell to form the resulting binary code from the values of the states of the corresponding cells, then we can obtain a two-dimensional array of binary or decimal codes.

In [15], a new form of evolution is presented, which consists in the fact that the cells of each state of the TCA at one coordinate form a code that corresponds to a given color in the RGB format. A

three-dimensional cellular automaton is formed, the structure of which is presented in [15]. Such a structure can be implemented in hardware and can also be used to generate color images. Thus, evolution is represented by one two-dimensional cellular array of numbers, represented by a binary code and encoding the colors in each cell. An example of such coding is shown in Figure 4.

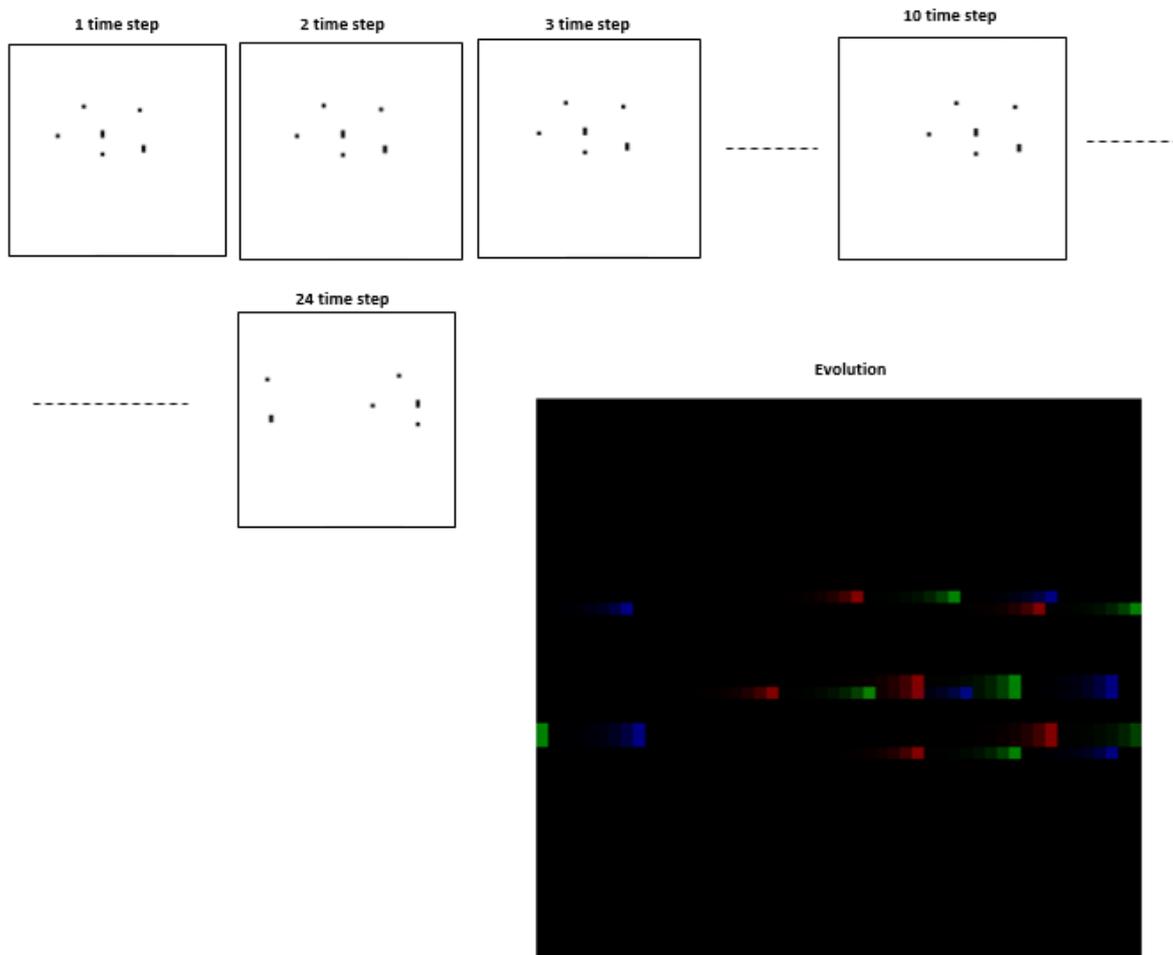


Figure 4: An example of TCA evolution on one two-dimensional array

Figure 4 depicts the process of changing states at 24 time steps, as well as on one two-dimensional array in color. Analysis of the slices shows that at a certain time step, it is possible to obtain the required binary image. In Figure 4 shows the evolution of a TCA in which the von Neumann neighborhood is used without taking into account the intrinsic state of the analyzed cell. Only four cells of the neighborhood are analyzed, which makes it possible to implement $2^4=16$ combinations of binary sets. Therefore, for four cells it is possible to implement $2^{16}=62236$ rules of transitions of TKA cells based on Wolfram coding. How the cells are numbered in such a neighborhood is shown in the work [15]. The top cell of the neighborhood is selected first (the least significant bit in the code), and the further numbering of the neighborhood cells is carried out clockwise.

If you use five cells (the cell's own state is taken into account), then the number of transition rules increases to $2^{32}=4294967296$. This is a large number of rules that are very difficult to experimentally investigate. Moreover, for the vicinity of Moore, there were even more rules.

In addition, for a different number of analyzed cells, different rules can perform the same transition function. However, this approach limits image processing and transformation to 24 time steps.

Another approach to constructing another form of TCA evolution is proposed. This evolution allows to increase the number of time steps, which are limited by the maximum set amount of code. The decimal representation of the three byte code is used here. The return to the initial state begins when the maximum possible code 11111111111111111111 is reached (encodes white). In this

case, the codes in the cells at the previous steps are preserved. In this view, the length (the number of time steps of evolution) of evolution varies. If during the entire time of evolution a TCA cell has a state of logical "0", then in the final image in this cell there is a black color. In this case, there may be situations when different cellular positions can be represented by different codes. In one position, the code may already start from zero, and in the next cell, the code may not yet reach the last state. An example of such a view in Figure 5 is shown.

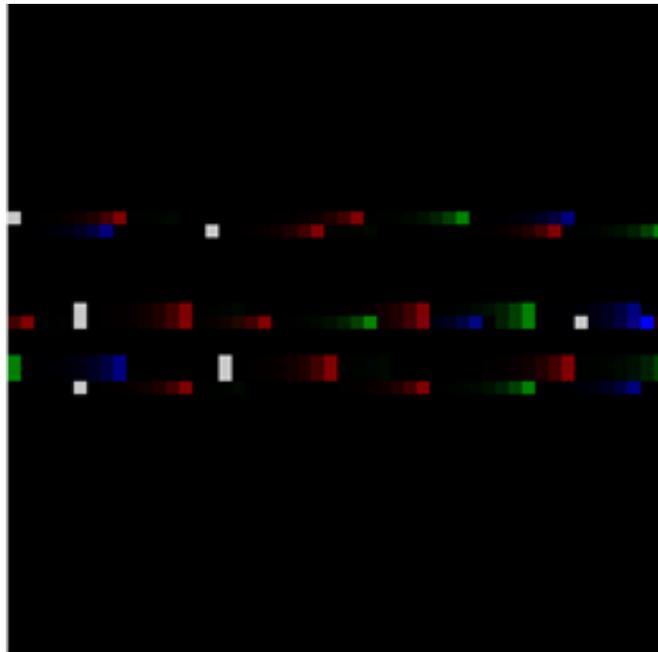


Figure 5: An example of evolution with decimal encoding when implementing rule 1000 (matrix size 50x50 pixels)

To display evolution (Figure 5), the rule 1000 is used, as well as four cells of the neighborhood without taking into account the own state of the analyzed cell. A two-dimensional array of cells after 44 time steps of TCA transitions is presented (matrix size 50x50 pixels). The initial state is used, shown in Figure 4. This rule implements a right shift of cells that have a single state and are located at a distance of two cells from each other, or two adjacent cells that are not diagonally located. If such an arrangement of cells is not supported, then rule 1000 does not implement a right shift for such an arrangement of cells. Here, at step 32, white cells appear (the maximum value is reached) and the process begins anew at the next time step. However, the rest of the cells remain in their previous states. Visually, some pixels may not display color changes. However, their binary code in the least significant bits of each byte can change.

With this mode, it is very difficult to predict the structure of the image and the distribution of colors at a large number of time steps. However, it is possible to construct a specific color image or obtain the desired binary slice (the state of the TCA at the selected time step of the iteration). For this, both forms of representation of evolution can be useful. Selecting specific rules allows you to perform image preprocessing operations.

4. Realization of Simple Operations on Images Based on the Formation of TCA Evolution

The simplest operation on images is the operation of shifting in the selected direction. Previously, the rule 1000 was considered, which implements a shift to the right for a certain arrangement of single cells. If we complicate the initial image, then the two-dimensional picture takes on a completely different form for the rule 1000. However, the movement and distribution of colors will be present on the same line of the location of cells with a state of logical "1". Therefore, to find the true rule of

transitions when implementing a shift of any arrangement of single cells, the analysis of logical sets and the arrangement of units in each set is carried out. The rules can be chosen in such a way that the lines of the image will visually intersect, but in fact, the intersection points will be located at different binary levels (different time steps of evolution).

Choosing the right rules is straightforward. The main thing is to know the shape of the neighborhood and the order of numbering of the cells that make up the neighborhood. We use the numbering order described in the work earlier and assume that only four cells of the neighborhood are analyzed without taking into account their own state. In this case, the counting starts from 1, since the zero cell number is assigned to the cell, which, based on the rule, changes its state at the next iteration step in evolution. In this case, the rules that implement the shift down, right, left and up for one time step of the iteration in evolution are described in Table 2.

Table 2

Transition rules 43690, 65280, 52428, 61680 for TCA

The state of the neighborhood cells at time t at the next time step				Shear direction (rule)			
$a_4(t)$	$a_3(t)$	$a_2(t)$	$a_1(t)$	down (43690)	right (65280)	left (52428)	up (61680)
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	0	1	0
0	0	1	1	1	0	1	0
0	1	0	0	0	0	0	1
0	1	0	1	1	0	0	1
0	1	1	0	0	0	1	1
0	1	1	1	1	0	1	1
1	0	0	0	0	1	0	0
1	0	0	1	1	1	0	0
1	0	1	0	0	1	1	0
1	0	1	1	1	1	1	0
1	1	0	0	0	1	0	1
1	1	0	1	1	1	0	1
1	1	1	0	0	1	1	1
1	1	1	1	1	1	1	1

An example of TCA evolution when implementing the rules described in Table 2 for shifting down, left, right, and up in Figure 6 is shown.

In Figure 6 shows the initial binary image (the initial state of the TCA), as well as four images of the evolution of the TCA for the rules 43690, 65280, 52428, 61680. As the original binary image moves, the single cells move to a higher binary slice, which changes the cell's binary code in the RGB system. Therefore, in the shifted images, the color and code of the cell of the color image change, which indicates the dynamics in the evolution of TCA. When moving from one color byte to another, the first unit bits of the byte appear invisible, since they are encoded by the least significant bits of each color byte. However, the states of logical "1" are present on the corresponding binary slices. In this case, the shape of the shifted image does not change.

The evolution of a TCA can be implemented not by one rule, but by a sequence of several rules. These rules can be repeated over many time steps in a row, or they can change frequently. So in Figure 7 shows the consistent application of the rules presented in table 2. Figure 7 it can be seen that the initial shift is carried out downward, then the shift is carried out to the right, further up and at the top, the image is shifted to the left. Only four rules are used, which allow you to simulate the movement of a selected object in any direction on a discrete orthogonal coverage.

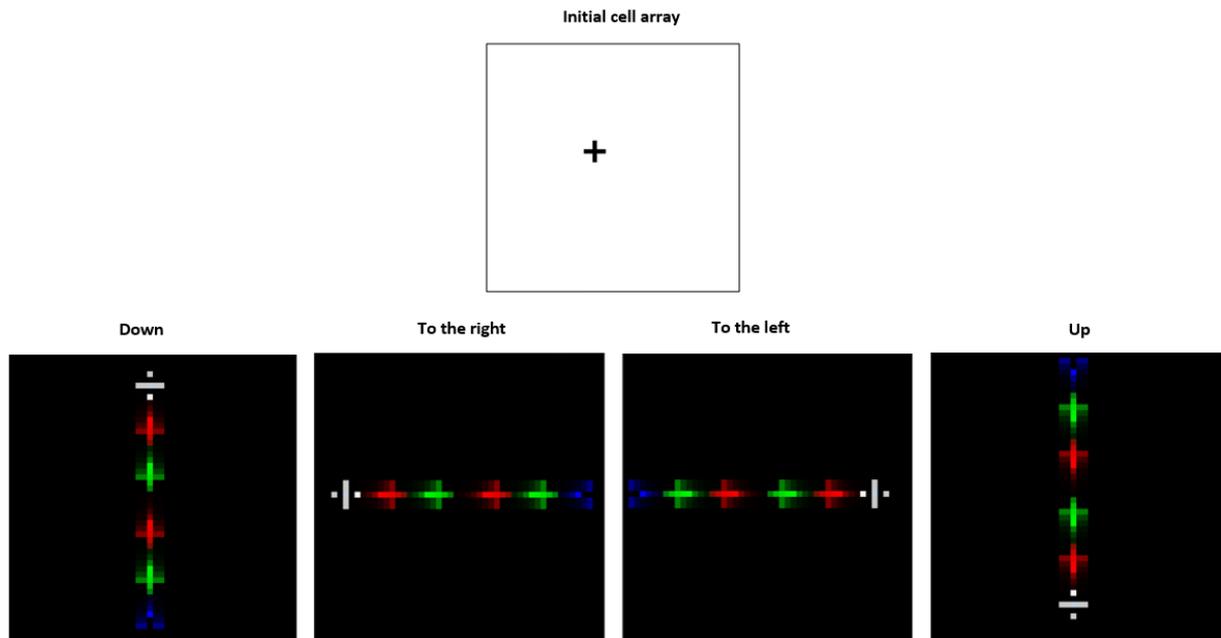


Figure 6: An example of applying the rules presented in Table 2. According to the initial state, the shifts are carried out down, right, left and down

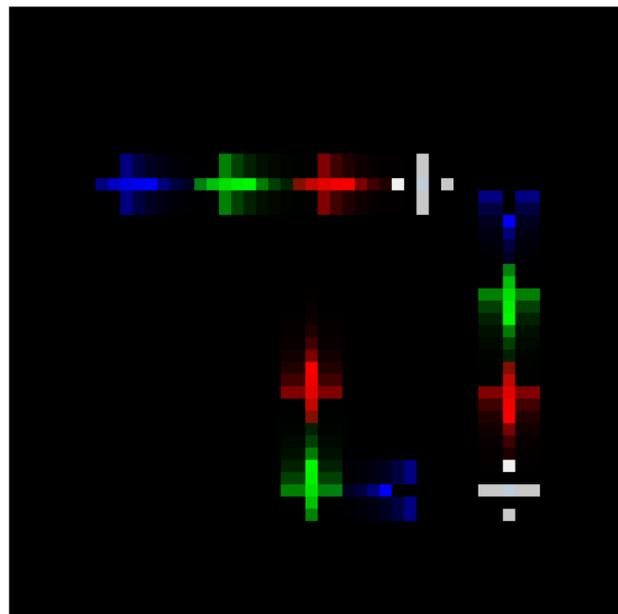


Figure 7: Evolution of TKA with the sequential application of four rules 43690, 65280, 61680, 52428 (the initial image consists of nine ones cells)

The choice of the appropriate slice allows you to fix any time moment of the location of single TCA cells. Enumeration of the rules allows you to build any trajectory, as well as an image of any complexity that is not trajectory. An example of such a trajectory in Figure 8 is shown. The trajectory is built sequentially, and accordingly, the color of each formed cell is changed. Accordingly, the evolution of DKA is traced. For the final formation of the trajectory, thresholding is used (Figure 9). In Figure 9 on the left, the trajectory is not represented by a solid line, since the coding of slices at low levels of TCA evolution weakly reflects colors. In this case, a single code in this binary slice is present.

possible to form color images of any complexity, are shown. Using a sequence of rules based on Wolfram coding simplifies imaging processes and simplifies their preprocessing operations. To solve the problem of image recognition, images are described by a sequence of rules that also form the base of standards. It is shown that for different forms of neighborhoods there are different rules that perform the same operation. This form of representation of the evolution of two-dimensional cellular automata makes it possible to simulate various dynamic processes in a field of the same dimensions. The implementation of such a structure by hardware is determined by the high speed of the formation of the image code in each cell, which depends on the switching time of the finite automaton of one cell in two-dimensional cellular automata. The modern element base allows achieving high switching speeds. The software implementation of this method requires sequential execution of all steps of the algorithm, which significantly consumes time resources. The most effective approach to the implementation of such structures is the use of FPGAs.

In future studies, the author plans to research and analyze, as well as the interaction of all rules for solving problems of describing and recognizing images.

6. References

- [1] Frank Y. Shih, *Image Processing and Pattern Recognition: Fundamentals and Techniques*, Wiley-IEEE Press, 2010.
- [2] Joe Minichino and Joseph Howse, *Learning OpenCV 3 Computer Vision with Python*, Packt Publishing - ebooks Account, 2015
- [3] Solomon C. and T. Breckon, *Fundamental of Digital Image Processing. A Practical Approach with Examples in Matlab*. Wiley – Blackwell, 2011.
- [4] Pratt W.K., *Digital Images Processing*. Third Gonzalez R.C. and Woods R.E., *Digital Image Processing*. 3rd ed., Prentice Hall, New Jersey. 2008
- [5] Bilan, Stepan, Elhoseny, Mohamed, & Hemanth, D. Jude (Eds.). (2020). *Biometric Identification Technologies Based on Modern Data Mining Methods*. Springer.
- [6] P. L. Rosin and X. Sun. Cellular Automata as a Tool for Image Processing. In: Chen, C. H. ed. *Emerging Topics in Computer Vision and its Applications*, Vol. 1. Series in Computer Vision, vol. 1. World Scientific Publishing, pp. 233-251.
- [7] Stepan Belan, Nikolay Belan. *Temporal-Impulse Description of Complex Image Based on Cellular Automata*, PaCT2013, LNCS, Vol. 7979, Springer-Verlag Berlin Heidelberg, (2013): 291-295
- [8] Wolfram, S. (2002). *A new kind of science*. Wolfram Media
- [9] Wolfram, S. (1983). *Statistical mechanics of cellular automata*. *Reviews of Modern Physics*, 55(3).
- [10] edition. Wiley, 2016
- [11] Mauri, Giancarlo, El Yacoubi, Samira, Dennunzio, Alberto, Nishinari, Katsuhiro, & Manzoni, Luca (Eds.). (2018). *Lecture Notes in Computer Science*. 13th International Conference on Cellular Automata for Research and Industry, ACRI 2018, Como, Italy. (September 17–21, 2018), Proceedings, 11115, Springer
- [12] ACRI. (2016). *Effects of Agents Fear, Desire and Knowledge on Their Success When Crossing a CA Based Highway*, at ABSim-CA Second International Workshop on Agent-Based Simulation & Cellular Automata, at the 12th International Conference on Cellular Automata for Research and Industry. ACRI 2016, Proceedings (September 05-08, 2016), Fez (Morocco), Sept. 05-08, 2016, Talk given on September 8.
- [13] Adamatzky, A. (2018). Cellular automata. A volume in the *Encyclopedia of complexity and systems science*. Second edition. Springer Science + business media LLC, part of springer Nature
- [14] Yuta, Kariyado, Camilo, Arevalo, & Julian, Villegas. (2021). Auralization of three-dimensional cellular automata. Romero et al. (Eds.). *EvoMUSART*, Springer, 161–170
- [15] Stepan Bilan. Evolution of two-dimensional cellular automata. New forms of presentation, *Ukrainian Journal of Information Technologies*, т. 3, №1, (2021): 85-90,