A novel method for raster image coding based on cellular automata evolution

Stepan Bilan[1], Ruslan Motornyuk[2], Oleg Ilarionov[1], Serhii Bilan[3]

[1]Taras Shevchenko National University of Kyiv, Volodymyrska Street, 60, Kyiv, 01033, Ukraine
[2]PU "Kiev Department" branch of the Main Information and Computing Center of the JSC "Ukrzaliznytsya", Ukraine
[3]Onseo company, Ukraine

Abstract

Nowadays, information technologies pay much attention to methods of efficient coding and compression of information. This is especially true for images. An important aspect is maintaining the clarity of the images with the least amount of visual distortion. The paper solves the problem of optimal coding of raster images based on the evolution of cellular automata. This approach makes it possible to preserve the confidentiality of the reproduced raster image, as well as to preserve the original raster image as a whole without visual distortion. The proposed method uses various forms of neighborhoods when forming the evolution of cellular automata, which makes it possible to reduce the time spent on implementing coding, and also reduces the amount of memory (number of bytes) used to store a raster image in encoded form in a digital computing system. Before encoding, the raster image is broken down into bit layers, which are also broken down into elementary fragments that are encoded. Evolutions of both elementary and two-dimensional cellular automata were used. All transition rules of elementary cellular automata, as well as two-dimensional cellular automata with von Neumann neighborhood, were considered. To achieve high accuracy and prevent losses during raster image reproduction, the method uses additional templates only in cases where it is impossible to avoid mismatches during the formation of cellular automata evolutions. All comparisons for mismatches are performed in parallel, and additional evolution or additional template can be used at any stage if this results in no mismatches. Templates implement changes to the cellular automaton using the XOR function. Typically, a template consists of all zeros, and logical "1"s are present only in cells where there are mismatches. The work does not describe additional methods of data compression after the implementation of evolutions and additional templates that are known and often used in various archivers.

Keywords

Raster image, cellular automaton evolution, coding, binary layer, transition rule, image compression

1. Introduction

If you go towards the storm, the clouds will disperse.

Modern information technologies are characterized by the widespread use of digitalization, processing, generation and image recognition technologies. In this regard, one of the key points is the efficient encoding of images for their digital representation. Modern methods of image coding are based on such basic characteristics as: speed, volume, complexity and accuracy.

The accuracy of image representation in a computer system determines the correspondence of all spatial and brightness parameters to the original image. Much attention in image coding is paid to the use of efficient image compression methods, which reduces the load on the computer system's memory and on the channels through which images are transmitted. The complexity and speed of image encoding and decoding determines the performance of a computing system. These two parameters are influential for various specialized real-time systems.

Currently, there are a large number of graphic file formats (BMP, JPEG, GIF, TIFF, PNG, etc.), which are formed using various encoding methods [1, 2]. The basic format for raster images is BMP,

[1]International Workshop on Computational Intelligence, co-located with the IV International Scientific Symposium "Intelligent Solutions" (IntSol-2025), May 01-05, 2025, Kyiv-Uzhhorod, Ukraine

* Corresponding author.
bstepan@ukr.net (S. Bilan); xehap0@gmail.com (R. Motoryuk); oilarionov@gmail.com (O. Ilarionov); belansoft@gmail.com (S. Bilan)
0000-0002-2978-5556 (S. Bilan); 0000-0003-4478-7020 (R. Motoryuk); 0000-0002-7435-3533 (O. Ilarionov); 0009-0000-8370-2849 (S. Bilan)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-4035/Paper2.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

which uses the RGB color model. In the RGB model, the color characteristics of each pixel are encoded in three bytes, each of which forms a color gamut of red (R), green (G), and blue (B) colors. In total, 24 bits are used to encode 1677216 possible colors. BMP files take up the largest amount of memory, and the format itself is the base for existing graphic formats.

There are methods for converting raster images from one graphic format to another. However, information loss is often possible during such transformations. An important aspect is that it determines the visual distortions of the displayed raster images when they are reproduced on the screen.

All coding methods use open visual data that is explicitly stored in the memory of the computer system. However, all known methods do not use raster image encoding implicitly. For example, such methods do not use a specific initial fragment of the image to unfold it into a full image. This approach also allows storing and transmitting a raster image in encrypted form. To implement such methods, various group operators can be used [3], and the most effective methods are those using cellular automata (CA) technologies.

2. Statement of the problem

Modern methods of encoding raster images do not always meet the needs of modern information technologies and users of computer systems and networks. This is especially relevant in terms of combining confidentiality and ease of encoding raster images. At the same time, it is important to reduce the volume when forming graphic formats in computer systems.

To solve the described problems, the paper sets the task of efficient coding of raster images based on the evolution of cellular automata, which allows describing images using various rules of evolution of one-dimensional and two-dimensional cellular automata. Using a sequence of cellular automata evolution rules allows to preserve the confidentiality of graphic information and also reduce the size of the raster image itself.

3. Relative works

Most raster image coding methods are considered from the point of view of data volume reduction [3, 5]. An important aspect is the order in which the bitmap is displayed on the screen. In this regard, various difficulties arise in the reproduction of raster images, which are caused by the coding and compression methods used [6]. Basically, all the most common methods of encoding raster images are considered in various file formats (BMP, JPEG, GIF, TIFF, PNG). These graphic formats implement such methods as:

- Run Length Evaluation (RLE) [7, 8].
- LZW encoding method [9].
- Huffman coding method [10].
- JPEG method [11, 12].

There are also many other formats for storing graphic data, with which the original image (BMP format) can be reproduced with partial loss of data or without loss. At the same time, these methods require significant computing resources for their implementation. The more a method compresses data, the more computational resources it requires to do so.

The well-known VQ method of signal compression has a major drawback, which determines the coding speed [13] The large dimensionality makes searching for a suitable template in the codebook quite resource-intensive for practical use. To reduce computing resources in the VQ method, different approaches are used, one of which is using mixed pyramidal structures.

The search for efficient methods of image coding using adaptive coding based on deep convolutional neural networks is underway [14, 15]. However, this method uses different compression levels applicable to JPEG formats. Decisions are made based on modeling relationships that are not always precisely defined. In addition, the method is based on JPEG encoding and does not have its own basis that distinguishes it from other image encoding methods. Basically, all raster image coding methods are based on the use of convolutional neural networks and are aimed at improving known coding methods for further image processing [15].

Codebook-based coding is also used, which uses semantics, which improves image reconstruction and classification accuracy [16]. At the same time, there is a need to form a semantic code book, which entails inaccurate results. Especially if training and artificial neural networks are additionally used.

There are also image coding methods that combine compression sensing and chaos theory, as well as signal sparsification in the wavelet transform domain [17]. The use of several algorithms with large computing resources often leads to various kinds of distortions of the reproduced images. The presented method is also quite effective for grayscale raster images.

Many works pay attention to image coding based on CA [18-21]. In the work [18] a method of coding raster images based on reversible CA as nonlinear filters is defined. This approach is suitable for compressing binary grayscale images. The method outperforms JPEG for smooth raster images and requires improvement for more complex images.

CAs are often used for efficient image encryption [21, 22]. In this plan, with the help of the CA, the code of each pixel of the image is changed, but the initial encoding of the image is not performed. The original raster image format is preserved.

Neural CAs (NCAs) [23] together with variational encoders (VAEs) are used for image coding and generation. In this paper, NCA is used as an additional tool to improve the accuracy of image generation to some extent.

Also, CAs are the basis for coding raster images using LDPC codes [20]. LDPC matrices are structured according to non-unified CA rules. However, this approach does not detect all errors due to low density, and is also characterized by high decoding complexity.

The work [24] presents research in the field of application of CA for image construction. CAs with different numbers of states for image construction are considered. However, the construction of color images is not described and the problems that arise in this regard are not investigated.

The works [3, 25, 26] consider the issues of dividing a raster image into binary layers and using each binary layer as a CA, which, according to specified transition rules, transforms each layer and the image as a whole. In these works, CAs are used to transform and generate raster images. Based on the described rules and the evolution of the CA, the movement of various geometric shapes is described, as well as the transformation of previously formed images. However, these works do not describe the process of initial encoding and description of images.

The proposed work allows one to initially formulate a sequence of actions for generating raster images in terms of the evolution of the CA.

4. Evolution of cellular automata in raster images

The use of CA for the formation of raster images is well described in [3, 25, 26]. They show the multilayer structure of an RGB image. Such an image is divided into 24 binary layers, each of which consists of bits of the same levels in the pixel codes. Since the code of each pixel consists of 24 bits (3 bytes), the codes of all pixels in the image form 24 two-dimensional bit arrays. Each of these layers can be considered as a cellular automaton. If we use a bit layer consisting of the least significant bits of the codes of all image pixels as a CA with initial states, then we can form a three-dimensional bit structure based on the evolution of the CA. To do this, it is necessary to specify the shape of the neighborhood and select a rule that is determined by the local state function for each cell of the CA [27]. An example of the evolution of an elementary CA (ECA) in Fig. 1 is shown.

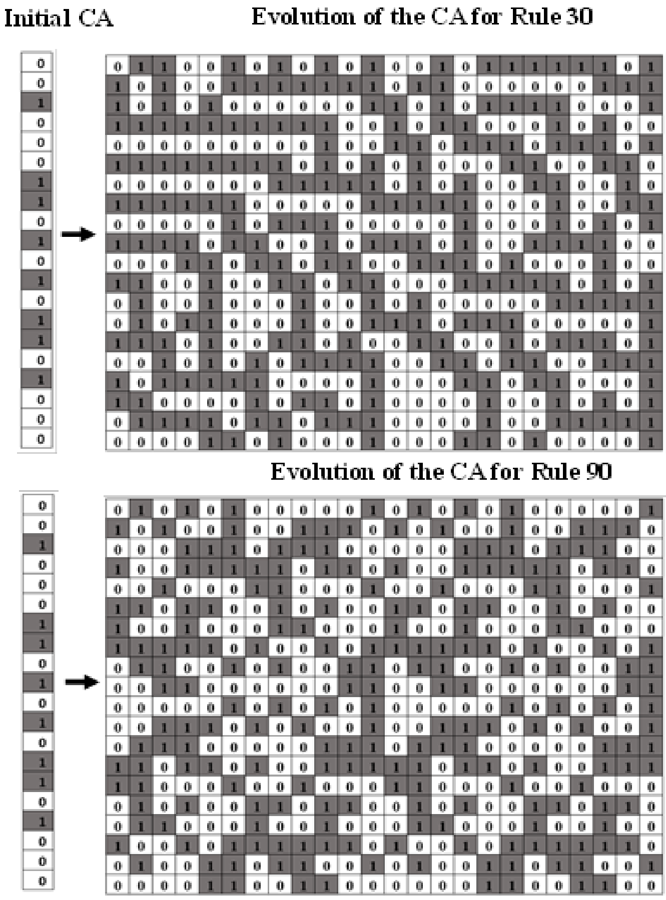

Figure 1: Example of ECA evolutions for rules 30 and 90

In Fig. 1, two rules of 30 and 90 transitions of the CA are used, with the help of which the initial states of the ECA passed 24 evolutionary steps. In fact, at each evolutionary step, an ECA is formed, the states of the cells of which depend on the states of the ECA cells at the previous step of evolution, as the corresponding binary layer of the image, then it is possible to obtain the necessary color raster image. In accordance with the considered example (Fig. 1), two images were generated for two rules, shown in Fig. 2.

Figure 2: The generated Images according to the ECA evolutions that indicated in Fig. 1.

Based on this approach, a raster image can be generated by selecting the initial states of the CA and specifying a sequence of evolutionary rules. In this case, the information will be open only to the initial bit layer. The remaining data will be presented implicitly using the sequence of evolutionary rules of the CA.

The considered examples (Fig. 1, Fig. 2) can use 256 evolutionary rules, since the arguments of the local state function for each cell are the state signals of its own cell and two neighboring cells. Such a limited number of rules does not always make it possible to form an image using these rules. Therefore, additional other forms of neighborhoods are used, which allow us to form the necessary basis of evolutions for the formation of the entire spectrum of raster images.

For two-dimensional CAs (DCAs), neighborhoods consisting of a larger number of cells are used. For the von Neumann neighborhood, four cells are used, which defines 2^{16} rules for the evolution of such DCAs. If we also use our own control cell, the number of evolutionary rules increases to 2^{32}. This makes it much more difficult to fully explore all the evolutionary rules in terms of generating raster images. In works [25, 26] evolutionary rules are considered that allow one to form the dynamics of an object’s movement along a movement field. Fig. 3 shows an example of image formation based on functions of one and two zeros.

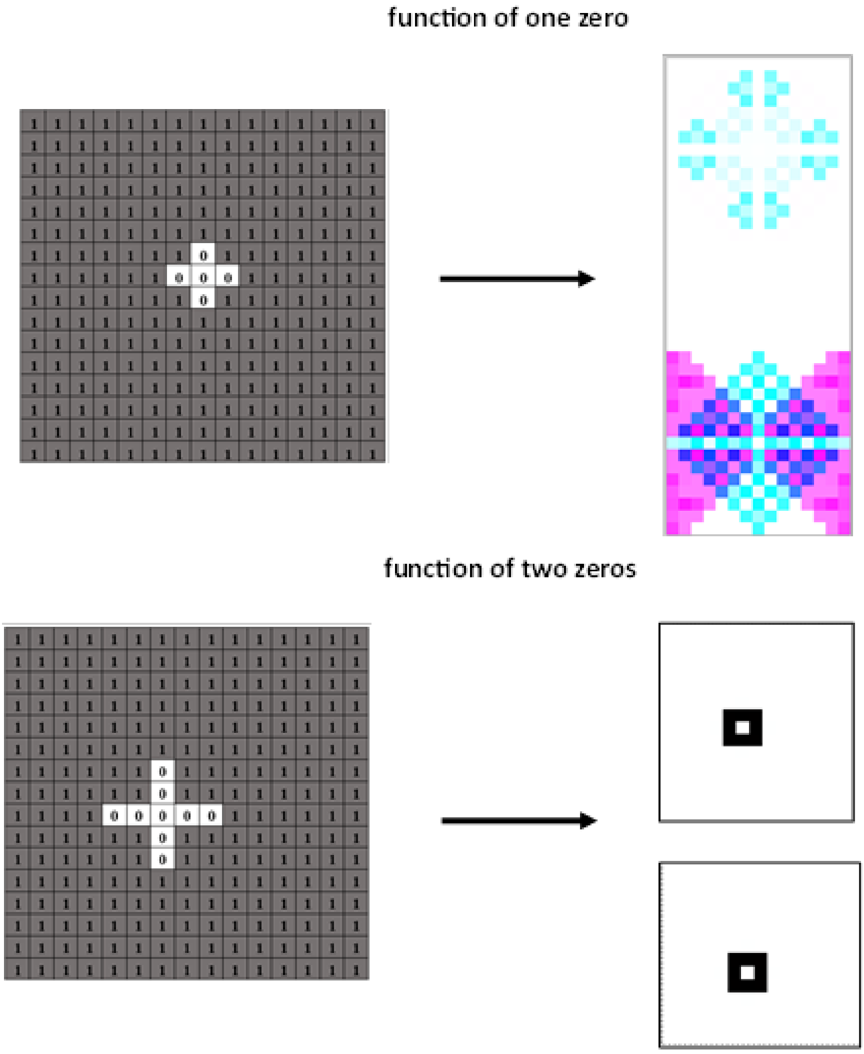

Figure 3: Example of image formation based on functions of one and two zeros

Figure 3 shows the initial bit arrays and images formed as a result of applying the one- and two-zero functions. The one-zero function is defined by the fact that only one cell in the neighborhood is in the state of logical "0". This circumstance, at the next evolutionary step, transfers the control cell to the state of logical “0”. The same approach is applied for the two zeros function, etc.

The one-zero function is described by the following formula.

$$b_{01}(t+1)=\begin{cases}0, if \sum_{i=1}^{4} a_i(t)=3 \\ 1, \in other\ cases\end{cases}, \quad (1)$$

where $b_{01}(t+1)$- the state of the control cell at the *(t+1)*th evolutionary step;

$a_i(t)$ – *i*-th cell of the von Neumann neighborhood for the cell $b_i(t)$.

The function of two zeros is described by a similar formula with a slight difference in the value of the sum.

$$b_{02}(t+1)=\begin{cases}0, if \sum_{i=1}^{4} a_i(t)=2 \\ 1, \in other\ cases\end{cases}. \quad (2)$$

Figure 3 shows two variants of the organization of the DCA. The first variant (the upper part of the resulting images) uses a CA in which the outer cells are not adjacent. In the second variant, the spread of state transfer is not limited to edge cells.

The analysis of the evolution of the CA and the experiments conducted allowed us to create a method for encoding raster images based on the evolution of the CA.

5. Method of coding raster images based on the evolution of the CA

Based on the described theoretical principles, a method for encoding raster images has been developed that uses the evolution of the CA. For this purpose, the raster image is divided into one-dimensional sections, each of which is encoded separately. Since a raster image is a two-dimensional matrix array of numbers, either vertical columns or horizontal rows can be selected as one-dimensional areas. The selected parts are split into binary layers. From these binary layers, a layer is selected (usually the layer consisting of the least significant bits of the codes of each pixel), which is the initial one. This initial layer evolves as an ECA and, in the process of evolution, an ECA is selected, formed at one of the evolutionary steps, at which the states of the ECA coincide with the states of the ECA that determines the next binary layer of the raster image element.

Similar processes are carried out with the subsequent binary layer of the raster image until a complete evolutionary analysis of all binary layers is performed. In this case, a fragment of a raster image is described by a sequence of rules and evolutionary steps that encode this fragment. The description of a raster image fragment begins with an initial array of numbers that represents its initial binary layer. The following describes the sequence of rules and numbers of evolutionary steps that, based on these rules, give complete matches with the next bit layer. The most ideal and simple notation of the image fragment code is presented in Table 1.

Table 1
Representation of the image fragment code based on the evolution of the CA

Initial bitmap fragment of a raster image	Bit layers. Rules of F_i and numbers of N_i evolutionary steps				
0th bit layer	1st bit layer	2nd bit layer	3rd bit layer	…………	23rd bit layer
[S]	(F_1, N_1)	(F_2, N_2)	(F_3, N_3)		(F_{23}, N_{23})

Table 1 shows the ideal option for encoding each subsequent bit layer. However, in real images it is not often possible to encode the subsequent bit layer with one rule. Especially if one of the layers contains all zeros or ones. To solve this problem, additional operations are used, information about which is entered into the general description of the raster image fragment.

One solution is to use additional evolutions that are formed based on the neighborhoods that other cells form. For example, two right or two left cells. Also, one option is to impose templates using the XOR function on a cellular automaton formed at one of the evolutionary steps and having the least number of differences with the subsequent binary layer. As a rule, the template has the dimension of the corresponding ECA and consists of all logical "0". Logical "1" states are present only in those template cells in which mismatches are detected. Mismatches are determined by the evolutions of all ECA rules. An example of visual identification of discrepancies for two ECAs in Fig. 4 is shown.

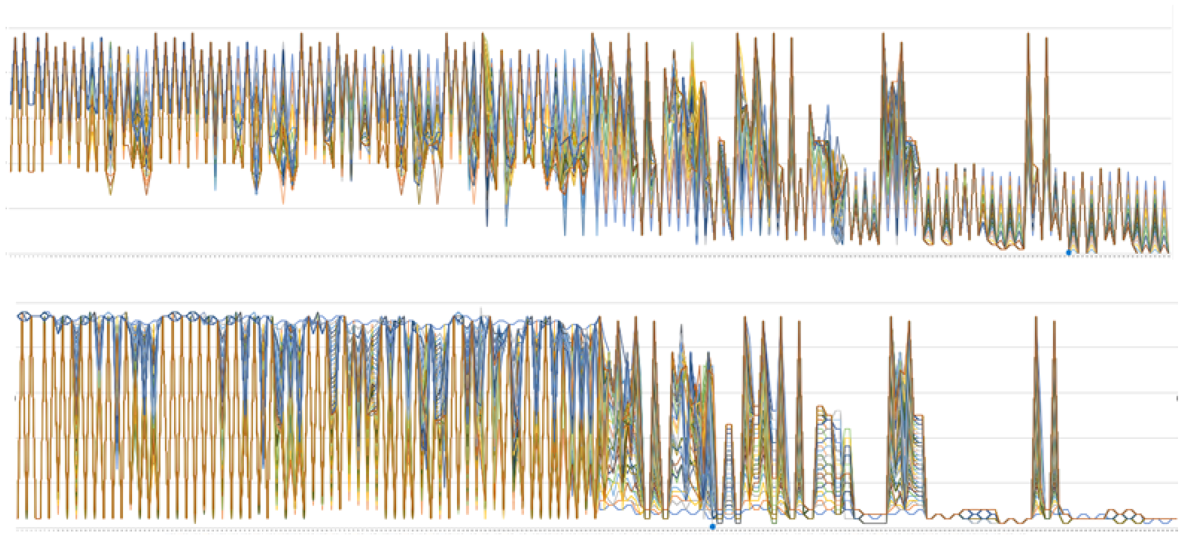

Figure 4: Examples of visual identification of mismatches for the previous and subsequent ECA

The graphs show at which evolutionary step there is a complete match (the vertical value is zero) and at which evolutionary step there are mismatches (the number indicating the number of mismatches). A rule (F_i) and an evolutionary step (N_i) are selected at which the smallest number of mismatches is observed (ideally, the number of mismatches should be equal to 0). If additional templates are used, the image is described in accordance with Table 2.

Table 2
Representation of the image fragment code based on the evolution of the CA and the use of additional templates

Initial bitmap fragment of a raster image	Bit layers. Rules of F_i and numbers of N_i evolutionary steps				
0th bit layer	1st bit layer	2nd bit layer	3rd bit layer	…………	23rd bit layer
[S]	(F_1, N_1) (z/o, M_l), $l=\overline{1,L_1}$	(F_2, N_2) (z/o, M_l), $l=\overline{1,L_2}$	(F_3, N_3) (z/o, M_l), $l=\overline{1,L_3}$		(F_{23}, N_{23}) (z/o, M_l), $l=\overline{1,L_{23}}$

In Table 2, the value of L_i determines the number of transitions between groups of zeros and ones in the additional pattern. From Table 2 it is evident that additional code combinations are added, which can form a common code that has a larger volume than in the BMP format. The use of ECA evolutions obtained from ECAs with other neighborhood shapes allows the need for additional binary templates to be minimized. Also reducing the amount of additional code is the use of the correct choice of template with the least number of logical "1"s, which reduces the number of transitions between groups of zeros and ones in the additional template.

An effective approach is to select the ECA with the least number of mismatches and transfer this ECA to the initial evolutionary stage. The ECA with the least number of mismatches obtained at a certain evolutionary step is subjected to evolutionary transformations and the number of mismatches is determined. This process continues until the number of mismatches is 0. If after such a procedure the number of mismatches does not decrease, then in this case an additional template is used, which completes the coding process.

In this case, the entry (F_1, N_1) will contain several rules and evolutions (F_{i1}, N_{i1}; F_{i2}, N_{i2},…, F_{ik}, N_{ik}). Here the value k determines the number of complete stages of evolutions carried out with ECA. The process of image coding based on the evolution of ECA is represented by the following sequence of steps.

1. The initial image fragment is selected and its initial bit layer (ECA_i) is formed, and the subsequent bit layer (ECA_{i+1}) of the selected image fragment is selected.
2. Evolutions of ECA_i are formed for each rule.
3. In the process of forming $ECA_i(t)$ for each rule and at each evolutionary step, a comparison of all $ECA_i(t)$ with the subsequent ECA_{i+1} is carried out. The comparison is carried out in parallel for all $ECA_i(t)$ at each step t of the evolution for all rules.
4. During the comparison process, the $ECA_i(t)$ values are recorded, which show the smallest number of cell mismatches.
5. If at one of the evolutionary steps for one of the rules the absence (zero) of mismatches is recorded, then the coding process ends. The entry (F_i, N_i) is carried out according to tables 1 and 2, which indicates the evolutionary transition from the initial bit layer ($ЭКА_i$) to the next bit layer (ECA_{i+1}) of the selected image fragment.
6. If at all evolutionary steps and at all rules there is no $ECA_i(t)$ in which there are no mismatches, then the $ECA_i(t)$ that shows the least number of mismatches is selected.
7. For the selected $ECA_i(t)$ several operations are carried out at the sixth stage.
    7.1 Evolutions are formed and comparisons are made in accordance with stages 1-6.
    7.2 A template is selected, which, using the XOR function, forms $ECA_i(t)$ and makes it possible to obtain ECA_{i+1} after the next evolutions. In this case, there should be no discrepancies.
8. After the operations carried out at stage 7, at the eighth stage, the results obtained at stages 7.1 and 7.2 are compared and the option that gives the least number of discrepancies and also has the shortest description according to table 2 is selected.

All further operations are carried out taking into account the described stages. These operations are performed for each subsequent bit layer. For large raster images, it is possible to set a certain number of mismatches that do not lead to significant visual distortions of the generated raster image.

As an example, we consider a 48×49 pixel image that represents the logo of an organization that conducts research in the field of information technology and cellular automata. The original BMP format of such an image in Fig. 5 is shown.

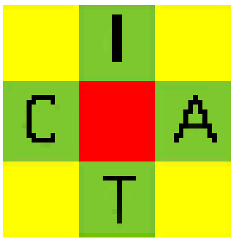

Figure 5: The original RGB image, which is encoded using the proposed method

To carry out the coding of the presented image, the ECA evolutions were used and only in that case additional templates were formed when the evolutions did not provide a complete absence of mismatches. 48 vertical columns were selected as separate fragments, which were divided into 24 binary layers, considered as ECA. An example of partitioning the 15th vertical column of the original image into binary layers in Fig. 6 is shown. Fig. 6 also shows the coding for each binary layer according to the proposed method.

The horizontal lines represent the 24-bit code of each pixel. In general, binary code represents the color of each pixel and can be represented as a decimal code. For example, for the fifth pixel from the top the decimal code is 65535, and for the 34th pixel the decimal code is 1112312. The entire raster image is a two-dimensional array of decimal numbers, each of which encodes a corresponding color.

For all 48 vertical columns, similar operations were carried out, as described in Fig. 6. The process of forming a description for a given raster image in Fig. 7 is shown.

Bit layers of the 15th column																							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	1
0	0	1	0	1	1	1	0	1	1	0	0	1	0	1	1	0	1	1	1	1	0	1	0
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	1
0	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 6: The bit layers of the 15th vertical column of the raster image shown in Fig. 5

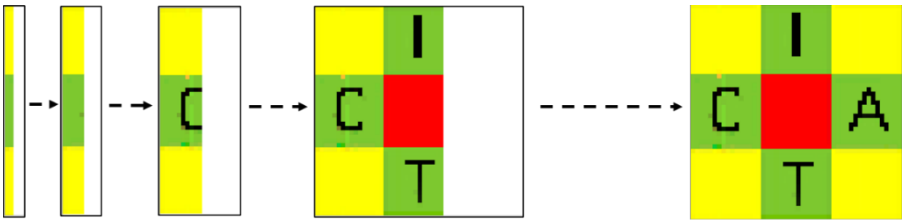

Figure 7: The process of forming and encoding each fragment of the original raster image

As can be seen from Figure 7, all fragments were formed taking into account the absence of mismatches. However, in Fig. 7, columns 6-8 are formed in such a way that coding was carried out when there were one or two mismatches. For such a small raster image, visual deviations are noticeable when it is enlarged. However, small deviations are hardly noticeable on large raster images.

An analysis of the amount of memory used to store this image was also conducted. It was taken into account that one byte is used to encode one rule, one evolutionary step, the number of zeros and the number of ones when using an additional template. For comparison, the BMP format was used, in which three bytes (RGB) were allocated for encoding one pixel. With this encoding, 147 bytes were allocated for one vertical column containing 49 pixels. Fig. 8 shows a graph of the differences in bytes for both encoding methods for each vertical column. A positive value indicates that fewer bytes are required for encoding using the proposed method compared to the BMP format.

From Figure 8 it can be seen that overall, a large number of bytes are freed for this image, equal to 2249 bytes. In this case, all the optimal coding methods described in this paper are not taken into account. In addition, no additional code compression methods were used. For example, series of zeros or ones and other numeric values were not searched. For large raster images, the compression effect of pure encoding is higher.

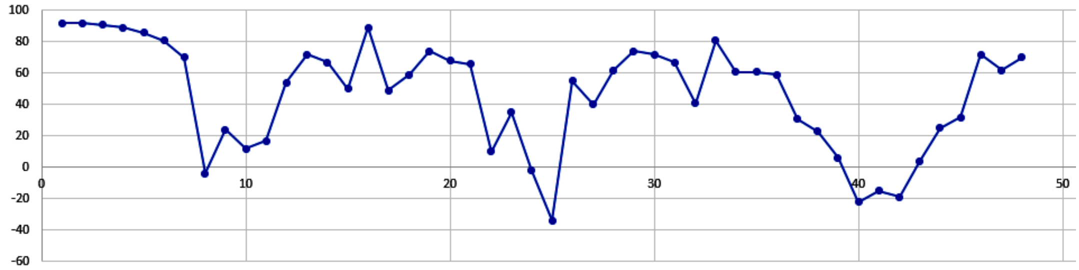

Figure 8: Graph of byte usage for image encoding in the proposed method compared to the BMP format

6. Conclusion

The paper considers a new method of encoding and describing raster images based on the evolution of cellular automata. This method allows one to describe an image in an implicit form. For this purpose, the operation of splitting the raster image into binary layers was used. Each of the binary layers does not carry complete information about the image. The use of binary fragments of raster images made it possible to preserve the confidentiality of the encoded image, since it is not described using color bytes, which contain all the information about the color and brightness of each point of the raster image.

Splitting a raster image into binary fragments that give complete matches with subsequent binary fragments has made it possible to reduce by several orders of magnitude the amount of memory allocated for storing raster images described using the evolution of cellular automata. Also, the volume was reduced by using evolutions of cellular automata with different forms of neighborhoods, which are formed in parallel and a simultaneous comparison of the resulting cellular automata is carried out, with a subsequent binary fragment of the raster image.

The method allows to preserve accuracy and reproduce the original raster image without loss of information by using additional templates in case it is impossible to eliminate all discrepancies by evolutionary means. This situation occurs when all cells of one of the binary fragments of the image are in the state of logical “0” or logical “1”.

Using this approach allows for the transmission of raster images in encrypted form and the encoding of information presented in three-dimensional form.

In further studies, the authors plan to conduct research in the direction of finding an optimal algorithm for compressing the obtained codes based on the evolution of cellular automata.

7. Declaration on Generative AI

The authors have not employed any Generative AI tools.

8. References

[1] J. D. Murray, W. Van Ryper. Encyclopedia of Graphics File Formats: The Complete Reference on CD-ROM with Links to Internet Resources. O'Reilly Media; Subsequent edition (May 1, 1996). 1116 pages.

[2] J. X. Chen. Guide to Graphics Software Tools, 2009, Springer London: 559. doi.org/10.1007/978-1-84800-901-1

[3] S. Bilan. Operators for Edge Detection in an Image Based on Technologies of Cellular Automata, International Conference "Information Technology and Interactions" (IT&I-2022). Workshops Proceedings Kyiv, Ukraine, Vol. 3384, 2022, pp.142-150

[4] J. Miano. Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP, 1999, ACM Press/Addison-Wesley Publishing Co. 264 pp.

[5] D. Salomon Data Compression. Springer-Verlag London Limited. (2007). 1092 pp. doi.org/10.1007/978-1-84628-603-2

[6] Yun-Qing Shi, H. Sun. Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards, Third Edition. 2021, CRC Press; 3rd edition. 664.

[7] Teaching guide: Run-length encoding (RLE). AQA 2020. URL: https://filestore.aqa.org.uk/resources/computing/AQA-8525-TG-RLE.PDF
[8] Recommendation T.45 (02/00): Run-length colour encoding. International Telecommunication Union. 2000. Retrieved 2015-12-06. URL: http://www.itu.int/rec/T-REC-T.45
[9] T. Welch."A Technique for High-Performance Data Compression". Computer. **17**(6), (1984), pp. 8–19. doi: 10.1109/MC.1984.1659158.
[10] A. Makandar, R. Biradar. Medical Image Compression Using Huffman Coding for Tiff Images. Lecture Notes in Networks and Systems ((LNNS,volume 779)). Data Analytics and Learning Conference paper, (2024). pp 151–161. doi.org/10.1007/978-981-99-6346-1_13.
[11] G. Hudson, A. Léger, B. Niss, I. Sebestyén, J. Vaaben. "JPEG-1 standard 25 years: past, present, and future reasons for a success".*Journal of Electronic Imaging*. **27** (4), (31 August 2018), pp. 1. doi:10.1117/1.JEI.27.4.040901. S2CID 52164892.
[12] A. Shukla. Efficient and Optimized Usage of Various Image Formats (JPEG/ JPG vs PNG) in Applications. Journal of Mathematical & Computer Applications. Volume 2(3), 2023, pp. 1-3
[13] Z. Pan; K. Kotani; T. Ohmi. Fast encoding method for vector quantization based on a new mixed pyramid data structure [image coding applications]. Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. pp. 397-400
[14] H. Chen, X. He, C. An, T. Q.Nguyen. Adaptive image coding efficiency enhancement using deep convolutional neural networks. Information Sciences. Volume 524, 2020, pp. 298-317
[15] F. Liu, J. Xu, J. Yang, W. Wu. A Novel Multi-Type Image Coding Method Acting on Supervised Hierarchical Deep Spiking Convolutional Neural Networks for Image Classification. Cognitive Computation. Volume 17, 2025. Article number 9.
[16] H. Zhang, M. Tao, Y. Sun, K. B. Letaief. Improving Learning-Based Semantic Coding Efficiency for Image Transmission via Shared Semantic-Aware Codebook. IEEE Transactions on Communications. Volume: 73, Issue: 2, 2025. pp. 1217 - 1232.
[17] S. Patel, A. Vaish. Efficient image coding through compressive sensing and chaos theory. Multimedia Tools and Applications. Volume 82, 2023, pp. 33225–33243.
[18] L. Cappellari, S. Milani, C. Cruz-Reyes; G. Calvagno. Resolution Scalable Image Coding with Reversible Cellular Automata. IEEE Transactions on Image Processing, Volume: 20, Issue: 5, 2011, pp. 1461– 1468. doi.org/10.1109/TIP.2010.2090531.
[19] L. R. Cheerotha, P. K. Ameefa. Cellular Automata Technology in Scalable Color Image Coding. International Journal of Engineering Research in Computer Science and Engineering, Vol 6, Issue 7, 2019. pp. 22-25.
[20] M. Anbuselvi, P. Saravanan. Image coding using Cellular Automata based LDPC codes. International Journal of Computer Science and Network Security, Vol.17 No.11, 2017. pp. 146-150.
[21] O. Omran. W. Alexan. Cellular Automata, S-Box and DNA Coding Based SPN for Image Encryption. 2023. 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). pp. 1-6.
[22] W. Alexan, M. ElBeltagy, A. Aboshousha. RGB Image Encryption through Cellular Automata, S-Box and the Lorenz System. Symmetry. 2022. pp.34-39.
[23] M. Chen, Z. Wang. Image Generation with Neural Cellular Automatas. 2020. URL: https://arxiv.org/abs/2010.04949
[24] E. E. Titova. Image construction by cellular automata. 2015, URL: http://intsys.msu.ru/magazine/archive/v12(1-4)/titova-105-122.pdf
[25] S. Bilan. Evolution of two-dimensional cellular automata. New forms of presentation, - Ukrainian Journal of Information Technologies, т. 3, №1, 2021, pp. 85-90.
[26] S. Bilan. A Technique for Describing and Transforming Images Based on the Evolution of Cellular Automata, - Intelligent Solutions (Computational Intelligence & Decision Making Theory). Vol-3106, 2021, pp. 106-115.
[27] Wolfram, S. A new kind of science. Wolfram Media, 2002.