

Digital image scrambling based on elementary cellular automata

Abdel Latif Abu Dalhoum¹ · Alia Madain¹ ·
Hazem Hiary¹

Received: 22 January 2015 / Revised: 8 August 2015 / Accepted: 25 September 2015 /
Published online: 19 October 2015
© Springer Science+Business Media New York 2015

Abstract Image scrambling is the process of converting an image to an unintelligible format, mainly for security reasons. The scrambling is considered as a pre-process or a post-process of security related applications such as watermarking, information hiding, fingerprinting, and encryption. Cellular automata are parallel models of computation that prove an interesting concept where a simple configuration can lead to a complex behavior. Since there are a lot of parameters to configure, cellular automata have many types and these types differ in terms of complexity and behavior. Cellular automata were previously used in scrambling different types of multimedia, but only complex two-dimensional automata were explored. We propose a scheme where the simplest type of cellular automata is used that is the elementary type. We test the scrambling degree for different cellular automata rules that belong to classes three and four of Wolfram's classification which correspond to complex and chaotic behavior; we also check the effect of other parameters such as the number of generations and the boundary condition. Experimental results show that our proposed scheme outperforms other schemes based on cellular automata in terms of scrambling degree.

Keywords Image scrambling · Elementary cellular automata ·
Complex and chaotic rules · One-dimensional scrambling

1 Introduction

A century ago, scrambling was the only way to hide information by converting it to an unintelligible format. Today, the application of scrambling varied significantly; although it

✉ Hazem Hiary
hazemh@ju.edu.jo

¹ The University of Jordan, Amman, 11942, Jordan

is still used in the field of security, it is considered as a pre-process and/or post-process of watermarking, information hiding, fingerprinting, and encryption.

Many multimedia scrambling methods that are based on cellular automata (CA) were proposed in the literature, complex two-dimensional cellular automata was proposed as a method of scrambling both images and audio in [1] and [14] respectively. Chaotic two-dimensional cellular automata was proposed in [27] as a method to scramble images.

CA is a discrete model of computation that could be used to solve any computable task. Cellular automata are widely used in different applications, such as generating images and music in art, random number generation, pattern recognition, routing algorithms and games. The application of cellular automata in the area of digital image processing includes image enhancement, compression, encryption and watermarking [28].

We propose a scrambling method based on simple one-dimensional cellular automata combined with repetition. This type can also have many varieties, but we choose the elementary type which is the simplest cellular automata with a one-dimensional grid and a neighborhood of the nearest cells only.

This work seeks to find out how to best benefit from elementary CA characteristics in image scrambling, where different CA types were tested to see which one would lead to a higher scrambling degree. An elementary cellular automaton is a one-dimensional cellular automaton where there are two possible states (labeled 0 and 1) and the rule to determine the state of a cell in the next generation depends only on the current state of the cell and its two immediate neighbors. As such it is one of the simplest possible models of computation.

The remaining of this paper is organized as follows: in Section 2 we review related works; Section 3 covers essential cellular automata background information; Section 4 describes the scrambling degree measurement; Section 5 presents the proposed scrambling algorithm; Section 6 gives the analysis and discussion of the experimental results, and finally, in Section 7, we conclude the work done and discuss possible directions for future work.

2 Related work

Digital image scrambling is an effective and common tool used in image encryption and data hiding, whose main goal is to reorder and change the position of image pixels and break the relationship between adjacent pixels [15]. There are many scrambling methods available. They include methods based on Advanced Encryption Standard [5], Twice Interval Division [24], Cat Chaotic Mapping [33], Magic Cube [29], Arnold Transformation [12, 17], M-sequences [32] and algorithms based on combinations [13].

Scramblers are often used as part of a multimedia security algorithm, such cases apply when scrambling algorithms are used in data hiding [16] or in encryption [9, 11, 23, 30] or watermarking [21]. Some scrambling algorithms are for specific types of images, for example, the work done in [10] covers a scrambling scheme of 3D images.

Two-dimensional cellular automata was previously used in scrambling different types of multimedia, it was used in audio scrambling as in [2], [4], and [14], as well as image scrambling as in [1] and [27]. In [1] the complex behavior was compared to the chaotic behavior scheme proposed in [27] and the complex behavior gives a higher scrambling degree than the ones of chaotic behavior.

We propose a general scrambling method based on elementary cellular automata (CA). Elementary cellular automata has the advantage of being simpler than any other type of CA,

not only for the one-dimensional grid but also for its limited verities, as there is a small number of possible rules and the allowed neighborhood is as simple as two direct neighbors.

After applying the elementary CA we found out that some of the commonly used chaotic behavior elementary CA achieves better scrambling degree than those of the complex behavior of the same type which is different from the outcome given in previous research when the two-dimensional CA were used.

Dimensions seem to have an effect on the scrambling degree, even in models that are not dependent on the CA concept. For example, in [31] authors use a one-dimensional chaotic system for scrambling images that is dependent on seed maps, another example is the work done in audio scrambling and watermarking that focus mainly on the effect of changing the dimension [6–8].

3 Cellular automata

Cellular automata were originally proposed by Stanislaw Ulam and John von Neumann in the 1940s, as formal models of self-reproducing organisms [22]. What is interesting about cellular automata is its simplicity and power. CA is capable of complex behavior despite the simple computational method behind it.

In general, a CA consists of a certain number of identical cells, each of which can take a finite number of states. The cells are distributed in space in one or more dimensions. At every time step, all the cells update their states synchronously by applying rules (also called transition function). The rules use the state of the cell and the state of its neighbors as input to determine its next state. CA differ in dimension, possible states, neighborhood relationships and rules. Elementary cellular automata are the simplest class of one-dimensional cellular automata. It has two possible values for each cell (usually 0 or 1) and the rules depend only on the values of the nearest neighbor. Consequently, the evolution of an elementary cellular automaton can completely be described by a table of the possible combination of each cell and its neighbors (8 possible states). The elementary cellular automata are usually referenced based on this table as there are only 256 elementary cellular automata (2^8), each of which can be indexed with an 8-bit binary number [22].

Figure 1 shows the way to calculate the rule number in order to refer to a specific elementary CA, in every cell in the first row there are three binary numbers which refer to the cell and its two direct neighbors, so in rule 30 if the cell is 0 and its direct left neighbor is 0 and the direct right neighbor is 0 then the cell in the table shows (000) and yields a zero in the upcoming generation. The eight states for each rule are concatenated to give the rule number.

The behavior of the CA is based on the rules used, Wolfram [22] classified one-dimensional (1-D) CA into four broad categories: (i) *Class 1*: ordered behavior; (ii) *Class 2*: periodic behavior; (iii) *Class 3*: random or chaotic behavior; (iv) *Class 4*: complex behavior.

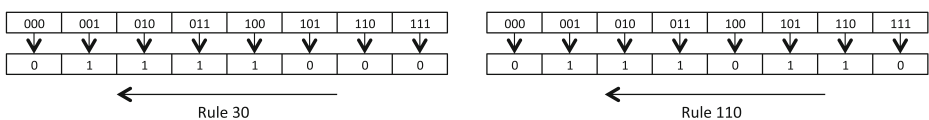


Fig. 1 Rule number

The first two are totally predictable. Random CA is unpredictable. Somewhere in between, in the transition from periodic to chaotic, the complex behavior occurs. In our experiments we will consider class three and four, since in class one and two nearly all initial patterns evolve quickly into stable or oscillating structures.

Another thing to consider is the grid boundary, in an infinite grid, every cell has a full neighborhood, but in a finite grid there must be a way to handle cells on the edges. A CA is said to have a null boundary if the left (right) neighbor of every leftmost (rightmost) cell is set to a zero state cell, and a periodic boundary if the extreme cells at opposite sides are adjacent to one another [18]. In our experiments, we will consider both periodic and null boundaries.

4 Scrambling degree

In order to compare our proposed scheme with previous schemes, we will use the gray difference concept. The main idea of using the gray difference is to calculate the correlation between adjacent pixels. If after applying the scrambling algorithm, the adjacent pixels are still related then the scrambling algorithm did not achieve its goal of breaking the correlation between adjacent pixels and achieving high diffusion and confusion properties. In [27] the scrambling degree is defined as follows:

- (1) Compute the gray difference between each pixel in the image and its neighboring pixels except the boundary.
- (2) Find the mean gray difference for the whole image except the boundary.

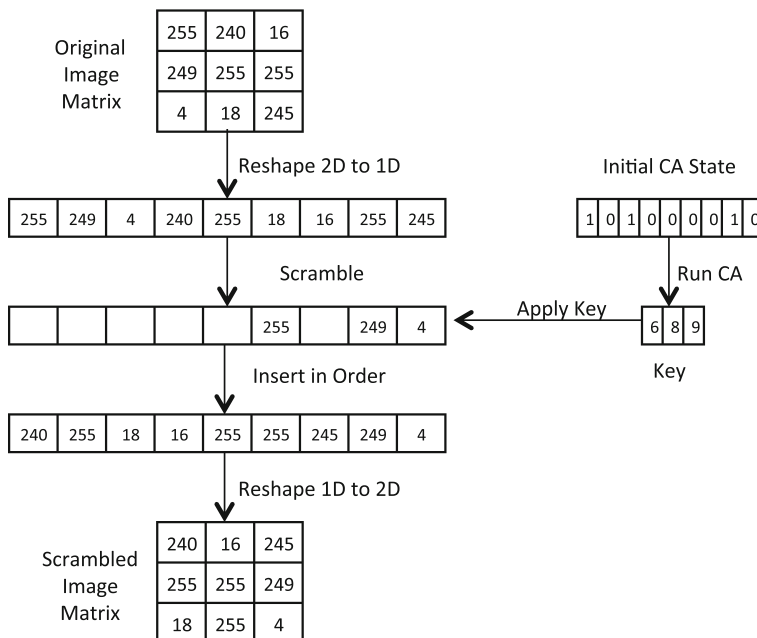


Fig. 2 Scrambling Algorithm Diagram

- (3) Find the scrambling degree from the mean gray difference of original and scrambled images.

Let $P(i, j)$ be the gray value of the pixel at position (i, j) and (\acute{i}, \acute{j}) denotes the coordinates $\{(i - 1, j), (i + 1, j), (i, j - 1), (i, j + 1)\}$. Equation 1 shows the calculation of the gray difference for each pixel in the image, the difference between the pixel and its neighboring pixels is squared and divided by 4.

$$GD(i, j) = \frac{1}{4} \sum_{\acute{i}, \acute{j}} [P(i, j) - P(\acute{i}, \acute{j})]^2 \tag{1}$$

After computing the gray difference for each pixel in the image –except the pixels at the boundary– the mean gray difference of the whole image can be computed by simply summing the gray difference and dividing over the grid size without the boundary as follows:

$$E(GD) = \frac{\sum_{i=2}^{M-1} \sum_{j=2}^{N-1} GD(i, j)}{(M - 2) \times (N - 2)} \tag{2}$$

Finally the scrambling degree is calculated, let $E(GD)$ be the mean gray difference of the original image and $\acute{E}(GD)$ be the mean gray difference of the scrambled image then the scrambling degree is calculated as follows:

$$GDD = \frac{\acute{E}(GD) - E(GD)}{\acute{E}(GD) + E(GD)} \tag{3}$$

Table 1 Image scrambling degree based on the number of generations

Image name	Number of Generations (NOG)			
	1	5	10	15
1.4.05	0.900	0.909	0.910	0.911
5.1.09	0.813	0.851	0.852	0.852
5.1.11	0.843	0.860	0.861	0.861
5.1.12	0.856	0.890	0.892	0.892
5.3.01	0.795	0.828	0.829	0.829
6.1.01	0.902	0.927	0.927	0.927
7.1.04	0.810	0.854	0.862	0.862
7.1.08	0.826	0.856	0.856	0.856
7.1.09	0.850	0.868	0.870	0.871
barbara	0.827	0.853	0.854	0.854
boat.512	0.771	0.809	0.811	0.812
camera	0.854	0.865	0.866	0.866
goldhill	0.866	0.889	0.890	0.891
gray21.512	0.988	0.989	0.989	0.989
mountain	0.796	0.819	0.820	0.820
peppers	0.835	0.860	0.862	0.862
texmos2.s512	0.874	0.889	0.892	0.892
texmos3.s512	0.950	0.955	0.955	0.955
zelda	0.880	0.901	0.902	0.902

The value of GDD will be a number between -1 and 1 . Better scrambling corresponds to an absolute value near to 1 .

5 Proposed algorithm

Scrambling refers to sufficient disorder of a semantic piece of media [25]. Figure 2 shows the diagram of the image scrambling scheme proposed. Adjacent pixels in an image have a strong correlation, and in order to scramble the image, this correlation needs to be reduced. The process starts with converting the image into a one-dimensional vector, afterwards the key generated by elementary CA is used to scramble the vector, and finally a reshaping operation is applied to convert the image back to its two-dimensional representation.

The algorithm starts by reading the input image and converting it to a one-dimensional array. This conversion is needed since the key produced by elementary cellular automata is one-dimensional. The key is generated by running the elementary CA to a number of generations, at each generation the position of cells that have the state of one are considered a part of the key and the cells that have the state of zero are ignored.

Assuming that the CA of the specified number of generations did not produce enough indices to cover all the original image pixels, these pixels are inserted in order, the reason why this works is that these pixels are inserted in the empty positions of the scrambled image which makes them scattered. In our experiments we show that repeating the usage of the same key more than once produces a higher scrambling degree. The reason why it is optional to repeat the process is that a good scrambling degree can be achieved without

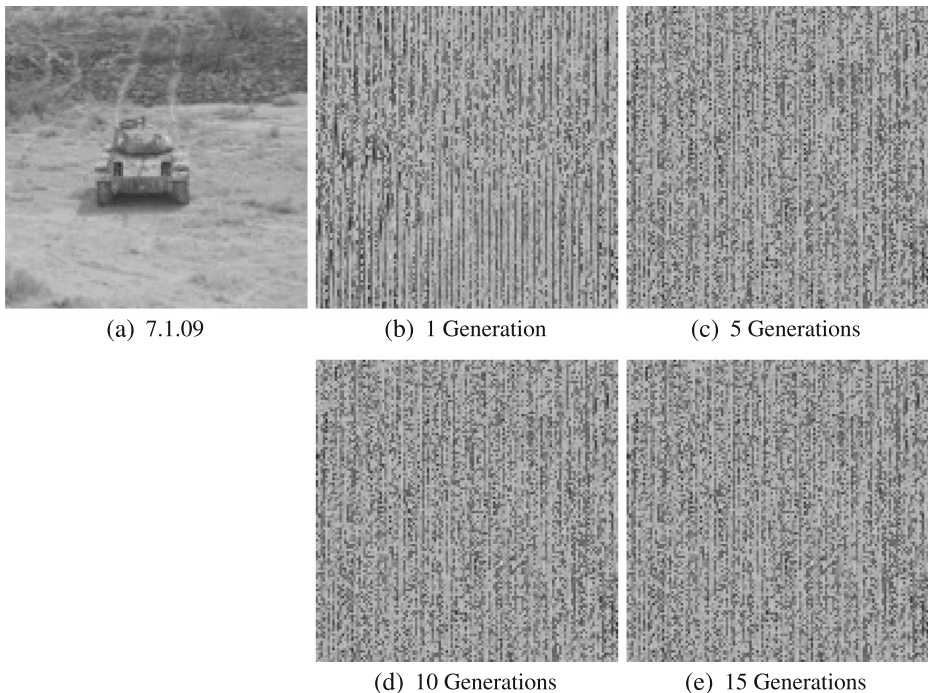


Fig. 3 7.1.09.tiff image scrambling using the proposed algorithm with different number of generations

any repetition. Finally, the one-dimensional array is converted back to its two-dimensional presentation.

The algorithm can be described as follows:

- Input** Original Image
Output Scrambled Image and key
Step 1 Read the input image and convert it to a one-dimensional array $A(x)$.
Step 2 Generate a scrambling key by running the elementary CA. In each generation get the indices where the cell state is one and add its position to the scrambling key.
Step 3 Read the key and modify $A(x)$, so that the first cell in the array is moved to the first index given by the key.
Step 4 If the key does not cover the original image dimension, insert the remaining pixels in order.
Step 5 Repeat steps 3 and 4, K times when needed.
Step 6 Reshape $A(x)$ back to the original image dimension. Note: if reshaping from two dimensions to one dimension in step 1 is column wise then reshaping back to two dimensions is also column wise.

In order to descramble the image, the process is reversed. There are two ways to reverse the process, the first method is to regenerate the key using the CA initial configuration and then apply it to the scrambled image, or to have the generated key available to descramble the image directly, without the need to regenerate the key.

Table 2 Scrambling degree when different boundary types are used

Image name	Boundary type	
	Periodic	Static
1.4.05	0.911	0.911
5.1.09	0.852	0.852
5.1.11	0.863	0.863
5.1.12	0.891	0.891
5.3.01	0.831	0.831
6.1.01	0.926	0.926
7.1.04	0.863	0.863
7.1.08	0.855	0.856
7.1.09	0.872	0.872
barbara	0.854	0.854
boat.512	0.812	0.811
camera	0.865	0.865
goldhill	0.889	0.889
gray21.512	0.989	0.989
mountain	0.823	0.823
peppers	0.862	0.862
texmos2.s512	0.893	0.893
texmos3.s512	0.955	0.955
zelda	0.900	0.900

6 Results and analysis

Unlike two-dimensional CA, the elementary ones does not need a neighborhood configuration as by definition elementary CA must depend on the nearest neighbors only. Other than getting the characteristics of CA evolving, the number of generations matter in this particular scheme as the scrambling algorithm depends on the result of each generation.

The data set used in these experiments is available at The University of Waterloo–Image repository [19] and USC-SIPI image database [20].

6.1 Number of generations

The number of generations is important as it gives the characteristics of the CA evolution. Also as our algorithm depends on these generations to generate the scrambling key it is important to study their effect.

We have tested the correlation between the scrambling degree and the number of generations by running the algorithm for the images-where each image is scrambled with the same key for the different generations, the results are shown in Table 1. The results range from 0.771 to 0.989.

In these tests periodic boundary, no repetition, same key per image, and rule number 22 were used. For the images we are using, the value of the scrambling degree does not differ

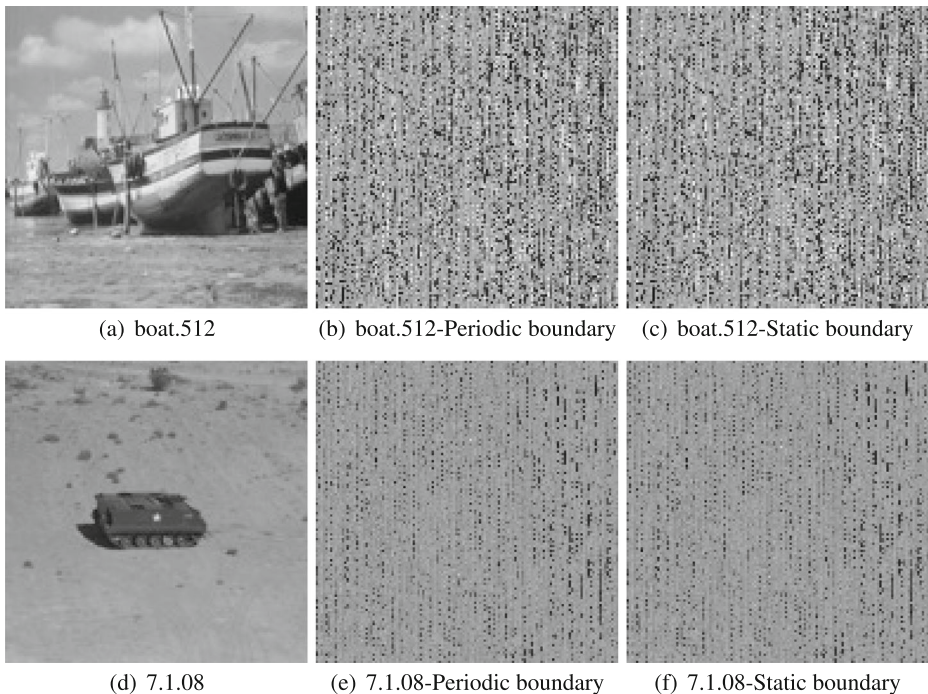


Fig. 4 Images scrambled with different boundary types

much after scrambling with ten generations so we will use ten generations in the following experiments.

The scrambling degree increases with the number of generations to some extent since this increase is bounded by the key used and the scrambling degree limits. In order to visualize the results, we show in Fig. 3 the result of scrambling 7.1.09.tiff for 1, 5, 10, and 15 generations. The results show that the image is hidden in all cases.

6.2 Boundary type

Work done in [1] combines neighborhood type experiments with boundary experiments. The conclusion was that the combination of Moore neighborhood and periodic boundary results in the highest scrambling degree. We retested the boundary effect as we are relying on elementary CA and a one-dimensional grid is likely to be different.

Boundaries effect more cells in a two-dimensional grid since it covers four sides of the grid and the number of pixels effected depends on the grid size. On the other hand, boundary adds exactly two cells in elementary CA. this is not to ignore the cumulative effect of these two cells as they will evolve and might affect all other grid cells. In order to test their effect each image was scrambled for 10 generations and the same key was used for each image, the results are given in Table 2.

Table 3 Scrambling degree with different chaotic rules and complex rule 110

Image name	Chaotic rule no.					Complex rule no.	
	30	22	150	126	182	110	
1.4.05	0.908	0.911	0.909	0.889	0.908	0.906	
5.1.09	0.847	0.850	0.848	0.802	0.841	0.832	
5.1.11	0.858	0.863	0.858	0.835	0.858	0.852	
5.1.12	0.890	0.890	0.889	0.847	0.884	0.888	
5.3.01	0.827	0.829	0.828	0.779	0.816	0.810	
6.1.01	0.926	0.927	0.926	0.897	0.917	0.919	
7.1.04	0.871	0.867	0.871	0.835	0.853	0.853	
7.1.08	0.851	0.855	0.844	0.814	0.840	0.829	
7.1.09	0.865	0.872	0.865	0.842	0.862	0.858	
barbara	0.852	0.857	0.850	0.807	0.841	0.839	
boat.512	0.809	0.813	0.812	0.771	0.803	0.804	
camera	0.863	0.862	0.863	0.803	0.853	0.844	
goldhill	0.887	0.890	0.886	0.866	0.879	0.882	
gray21.512	0.989	0.989	0.989	0.983	0.989	0.987	
mountain	0.822	0.824	0.823	0.778	0.811	0.806	
peppers	0.863	0.860	0.864	0.832	0.851	0.853	
texmos2.s512	0.888	0.893	0.885	0.863	0.877	0.876	
texmos3.s512	0.954	0.955	0.954	0.941	0.951	0.950	
zelda	0.894	0.901	0.894	0.873	0.886	0.889	

Figure 4 shows the result of scrambling boat512.tiff and 7.1.08.tiff, it can be seen from the images that the results are very similar.

6.3 Chaotic vs. Complex

Cellular automata can belong to different classes from repetitive or stable states to complex or even chaotic ones, the class of a certain CA depends on the rule used. In this subsection we will focus on rule 110 as an example of *Class 4* rules (complex rules) which has been shown to be capable of universal computation then we compare rules with chaotic behavior (22, 30, 126, 150, 182) with the rule 110.

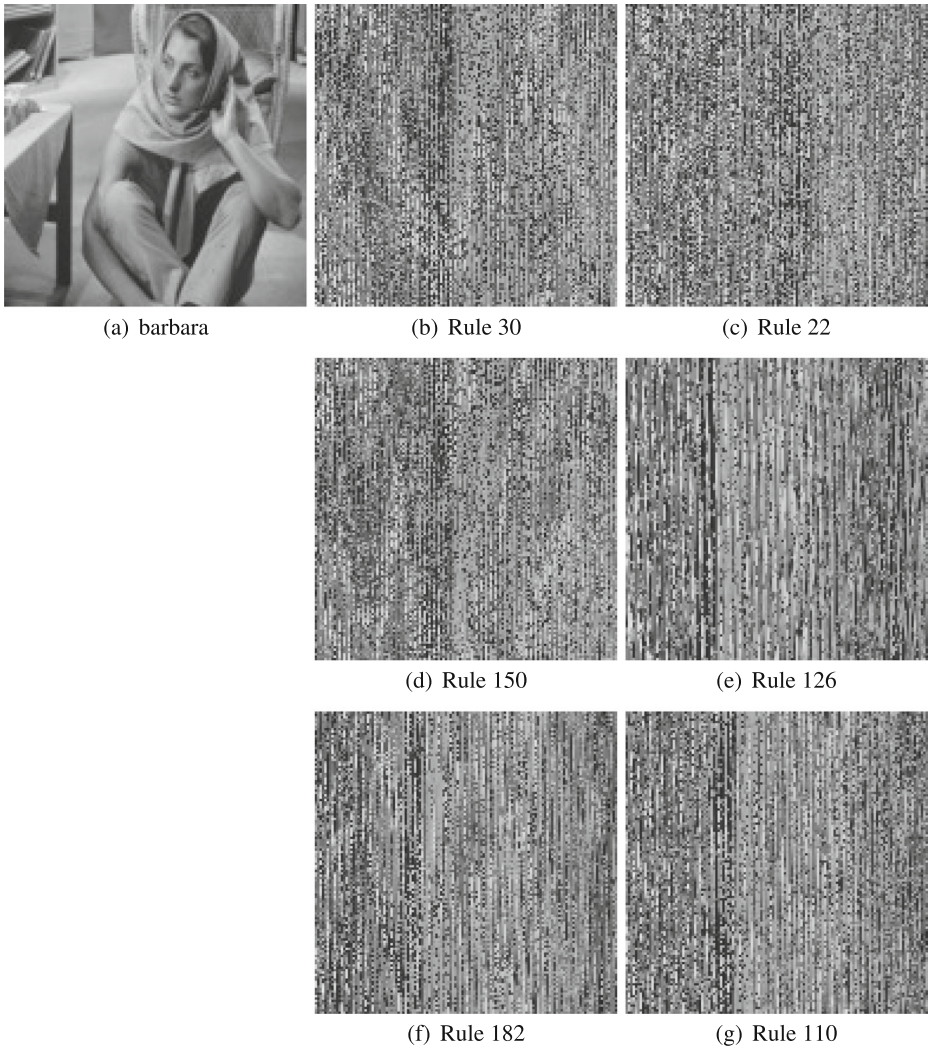


Fig. 5 barbara.tiff image scrambling based on different chaotic rules

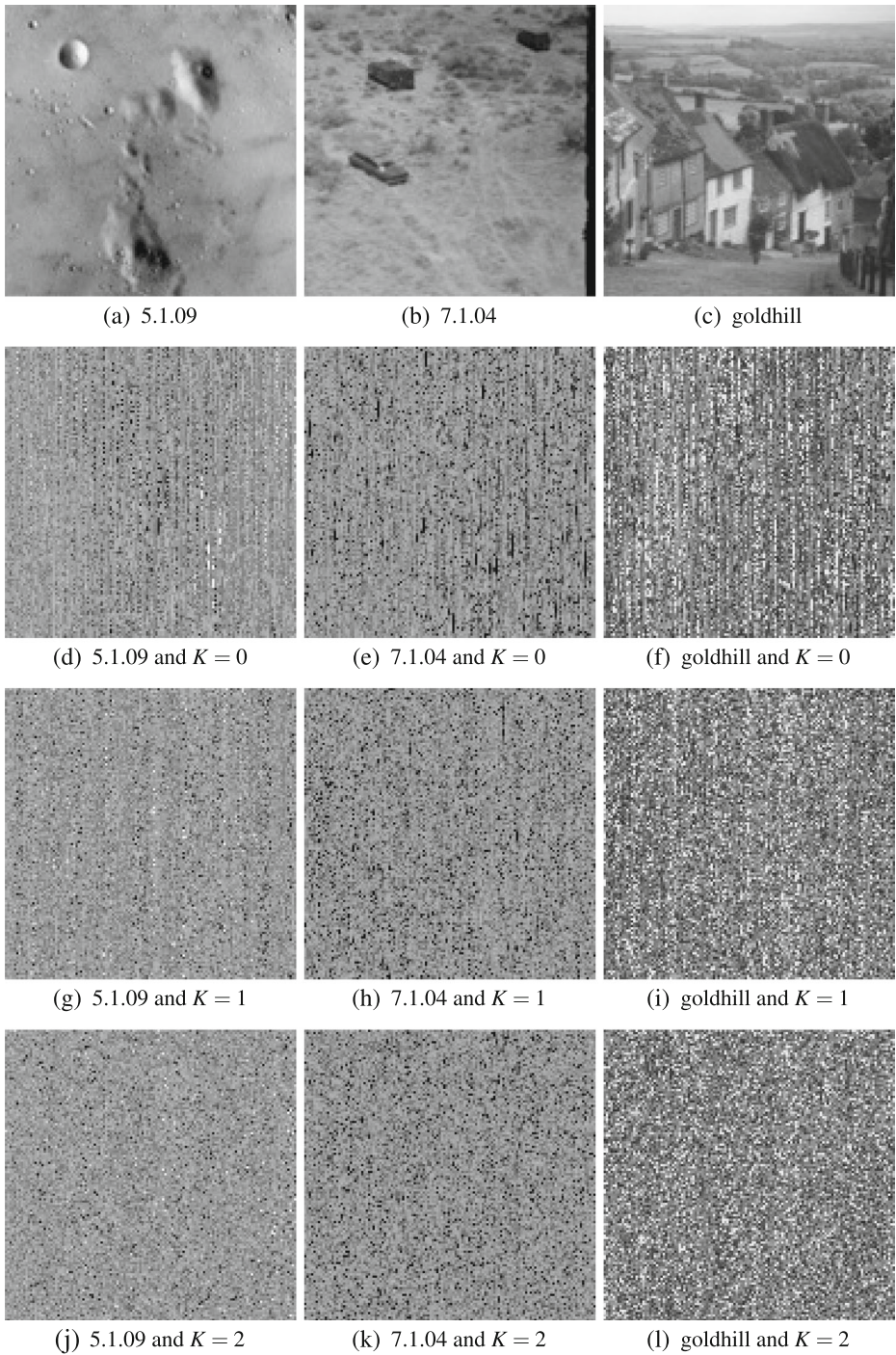


Fig. 6 Images scrambled with $K = 0; 1$ and 2

The results are quite different from those given in research based on two-dimensional cellular automata [1]. Table 3 shows the results of scrambling with different rule numbers, the same key is used for each image and the scrambling is done for 10 generations. It can be seen that some chaotic rules (not all of them) are better than the complex rule (rule 110) in terms of the scrambling degree. These rules are known in the field of random number generation so it was expected that they will provide good scrambling.

Figure 5 shows the results of scrambling barbara.tiff image using different rules. It can be seen that scrambling using some of the chaotic behavior rules achieve a higher scrambling degree than the rule 110 which exhibits a complex behavior.

6.4 Repetition effect

Until now we were using the scrambling algorithm without repetition which produces good results; in this section we will focus on the repetition effect. As shown in the algorithm the repetition is optional since the need for repetition is dependent on the application, in addition, the repetition given in the algorithm is dependent on one key only which is the same key used to scramble the image in the first place so no need to run cellular automata again to generate keys, the same key will be used for repetition.

Repeating the algorithm costs time and if repeated too much it might be for little or no gain, this case is similar to the one discussed in the number of generations section were repeating for too many generations will cost time and might give the same result. Figure 6 shows the repetition effect on scrambling for three images (5.1.09.tiff, 7.1.04.tiff

Table 4 Comparison between 2D scrambling scheme and our proposed scheme

Image	Scheme			
name	2D CA [1]	Proposed, $K = 0$	Proposed, $K = 1$	Proposed, $K = 2$
1.4.05	0.875	0.911	0.915	0.918
5.1.09	0.837	0.851	0.864	0.864
5.1.11	0.845	0.864	0.872	0.877
5.1.12	0.876	0.891	0.896	0.898
5.3.01	0.825	0.830	0.841	0.846
6.1.01	0.915	0.928	0.930	0.932
7.1.04	0.862	0.866	0.880	0.881
7.1.08	0.844	0.856	0.865	0.869
7.1.09	0.852	0.872	0.876	0.880
barbara	0.848	0.855	0.866	0.868
boat.512	0.803	0.812	0.824	0.826
camera	0.867	0.862	0.875	0.878
goldhill	0.884	0.889	0.898	0.900
gray21.512	0.989	0.989	0.990	0.991
mountain	0.816	0.821	0.832	0.835
peppers	0.860	0.864	0.875	0.877
texmos2.s512	0.891	0.893	0.899	0.901
texmos3.s512	0.951	0.955	0.958	0.959
zelda	0.889	0.900	0.908	0.909

and goldhill.tiff) when $K = 0$; 1; and 2. The more the scrambling is repeated the higher the scrambling degree. It can be seen from the figure that the repetition for two times looks more like salt and pepper noise.

6.5 Comparison

There are many scrambling methods in literature, in fact the complex two-dimensional cellular automata presented in [1] were tested against the Josephus algorithm (JA) [3] and the chaos mapping algorithm (LA) [26] and it achieved a higher scrambling degree, it was also tested against the chaotic CA and still it gives a higher scrambling degree.

Later, the two-dimensional CA was used in scrambling audio files, although these files are one-dimensional and using the two-dimensional CA in this context requires conversion from one dimension to the other. The proposed scheme points out that the simplest type of one-dimensional CA still can be used and the experimental results show that it achieves a higher scrambling degree than the others, especially when it is combined with repetition.

In order to compare with the 2D scheme presented in [1], we will use the same images and the same keys used in Section 6.4, the results are shown in Table 4. When no repetition is used the results are very similar to 2D CA but it shows higher scrambling with one or two repetitions, as in the case of the number of generations the benefit from repetition effect is bounded.

7 Conclusions and future work

This paper investigates the potential of using the simplest type of cellular automata in the field of image scrambling; the results show a good potential of this scheme to replace 2D cellular automata methods. We tested three important parameters of cellular automata, we first test the number of generations effect on scrambling, afterwards we test two different boundary types, namely the periodic and the static, and finally we test the rules. The experimental results show that 10 generations and rule 22 combined with any type of boundary result in a high scrambling degree compared with what can be achieved from a certain image. Compared with scrambling techniques based on 2D cellular automata, our scheme gives similar results without repetition and better results with repetition. The idea here is the simplicity of the elementary cellular automata and ability to test it thoroughly.

Future plans include applying this method to encryption, data hiding and watermarking applications. Testing extensively the whole set of elementary cellular automata rules (265 rule) is also left as a future work.

References

1. Abu Dalhoum A, Mahafzah B, Ayyal Awwad A, Aldhamari I, Ortega A, Alfonseca M (2012) Digital image scrambling using 2D cellular automata. *IEEE MultiMedia* 19(4):28–36
2. Augustine N., George S., Deepthi P. (2014) Sparse representation based audio scrambling using cellular automata. In: *IEEE International Conference on Electronics, Computing and Communication Technologies (IEEE CONECCT)*. IEEE, Bangalore, pp 1–5
3. Desheng X, Yueshan X (2005) Digital image scrambling based on josephus traversing. *Computer Eng and Applications* 10:44–46
4. George SN, Augustine N, Pattathil DP (2014) Audio security through compressive sampling and cellular automata. *Multimedia Tools and Applications*:1–25

5. Jiping N, Yongchuan Z, Zhihua H, Zuqiao Y (2008) A digital image scrambling method based on AES and error-correcting code. In: International Conference on Computer Science and Software Engineering (CSSE '08), vol 3. IEEE, Wuhan, pp 677–680
6. Li H, Qin Z (2009) Audio scrambling algorithm based on variable dimension space. In: International Conference on Industrial and Information Systems (IIS '09). IEEE, Haikou, pp 316–319
7. Li H, Qin Z, Shao L (2009) Audio watermarking pre-process algorithm. In: IEEE International Conference on e-Business Engineering (ICEBE '09). IEEE, Macau, pp 165–170
8. Li H, Qin Z, Shao L, Zhang S, Wang B (2009) Variable dimension space audio scrambling algorithm against MP3 compression. In: Hua A, Chang SL (eds) Algorithms and Architectures for Parallel Processing, Lecture Notes in Computer Science, vol 5574. Springer, Berlin, pp 866–876
9. Li H, Wang Y, Yan H, Li L, Li Q, Zhao X (2013) Double-image encryption by using chaos-based local pixel scrambling technique and gyration transform. *Opt Lasers Eng* 51(12):1327–1331
10. Li XW, Cho SJ, Kim ST (2014) A 3D image encryption technique using computer-generated integral imaging and cellular automata transform. *Optik - International Journal for Light and Electron Optics* 125(13):2983–2990
11. Liu S, Sheridan JT (2013) Optical encryption by combining image scrambling techniques in fractional fourier domains. *Opt Commun* 287:73–80
12. Liu Z, Gong M, Dou Y, Liu F, Lin S, Ahmad MA, Dai J, Liu S (2012) Double image encryption by using arnold transform and discrete fractional angular transform. *Opt Lasers Eng* 50(2):248–255
13. Liu Z, Zhang Y, Liu W, Meng F, Wu Q, Liu S (2013) A mixed scrambling operation for hiding image. *Optik - International Journal for Light and Electron Optics* 124(22):5391–5396
14. Madaïn A, Abu Dalhoum A, Hiary H, Ortega A, Alfonseca M (2014) Audio scrambling technique based on cellular automata. *Multimed Tools Appl* 71(3):1803–1822
15. Maleki F, Mohades A, Hashemi S, Shiri M (2008) An image encryption system by cellular automata with memory. In: Third International Conference on Availability, Reliability and Security (ARES '08). IEEE, Barcelona, pp 1266–1271
16. Parah SA, Sheikh JA, Hafiz AM, Bhat G (2014) Data hiding in scrambled images: A new double layer security data hiding technique. *Comput Electr Eng* 40(1):70–82
17. Shang Z, Ren H, Zhang J (2008) A block location scrambling algorithm of digital image based on arnold transformation. In: The 9th International Conference for Young Computer Scientists (ICYCS '08). IEEE, Hunan, pp 2942–2947
18. Shin SH, Yoo KY (2009) Analysis of 2-state, 3-neighborhood cellular automata rules for cryptographic pseudorandom number generation. In: International Conference on Computational Science and Engineering (CSE '09), vol 1. IEEE, Vancouver, pp 399–404
19. (2015) The University of Waterloo-Image repository:<http://links.uwaterloo.ca/Repository.html>. Accessed 12 Oct
20. (2015) USC-SIPI Image Database: <http://sipi.usc.edu/database/>. Accessed 12 Oct
21. Wang T, Li H (2014) A novel scrambling digital image watermarking algorithm based on contourlet transform. *Wuhan University Journal of Natural Sciences* 19(4):315–322
22. Wolfram S (2002) A New Kind of Science. Wolfram Media
23. Wu J, Guo F, Liang Y, Zhou N (2014) Triple color images encryption algorithm based on scrambling and the reality-preserving fractional discrete cosine transform. *Optik - International Journal for Light and Electron Optics* 125(16):4474–4479
24. Xiangdong L, Junxing Z, Jinhai Z, Xiqin H (2008) A new chaotic image scrambling algorithm based on dynamic twice interval-division. In: International Conference on Computer Science and Software Engineering, vol 3, Wuhan, pp 818–821
25. Yan W, Weir J (2010) Fundamentals of Media Security. Ventus Publishing Aps
26. Ye G, Huang X, Zhu C (2007) Image encryption algorithm of double scrambling based on ascii code of matrix element. In: International Conference on Computational Intelligence and Security. IEEE, Harbin, pp 843–847
27. Ye R, Li H (2008) A novel image scrambling and watermarking scheme based on cellular automata. In: International Symposium on Electronic Commerce and Security. IEEE, Guangzhou City, pp 938–941
28. Zefreh E, Rajae S, Farivary M (2011) Image security system using recursive cellular automata substitution and its parallelization. In: CSI International Symposium on Computer Science and Software Engineering (CSSE). IEEE, Tehran, pp 77–86
29. Zhang L, Ji S, Xie Y, Yuan Q, Wan Y, Bao G (2005) Principle of image encrypting algorithm based on magic cube transformation. In: Hao Y, Li J, Wang YP, Cheung YM, Yin H, Jiao L, Ma J, Jiao YC (eds) Computational Intelligence and Security, Lecture Notes in Computer Science, vol 3802. Springer, Berlin, pp 977–982

30. Zhong Z, Chang J, Shan M, Hao B (2012) Double image encryption using double pixel scrambling and random phase encoding. *Optics Communications* 285(5):584–588
31. Zhou Y, Bao L, Chen CP (2014) A new 1D chaotic system for image encryption. *Signal Process* 97:172–182
32. Zhou Y, Panetta K, Aгаian S (2008) An image scrambling algorithm using parameter bases M-sequences. In: *International Conference on Machine Learning and Cybernetics*, vol 7. IEEE, Kunming, pp 3695–3698
33. Zhu L, Li W, Liao L, Li H (2006) A novel algorithm for scrambling digital image based on cat chaotic mapping. In: *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '06)*. IEEE, Pasadena, pp 601–604



Abdel Latif Abu Dalhoom received his PhD degree in computer science from the University Autonoma De Madrid, Spain in 2004. He is currently an Associate Professor of evolutionary algorithms and complex systems at the University of Jordan. He has published about 25 papers in evolutionary algorithms, cellular automata, Fractals and DNA computing. He is a member of GHIA research group at the University Autonoma De Madrid.



Alia Madain is a PhD Candidate in the Department of Computer Science at the University of Jordan. She received her B.Sc. and M.Sc. degrees in Computer Science from the University of Jordan in 2009 and 2011 respectively. She has five years of work experience and her research interests are multimedia processing and security.



Hazem Hiary received his PhD in Computer Science / Image Processing from the University of Leeds (Leeds, UK) in 2008, B.Sc. and M.Sc. degrees in Computer Science from the University of Jordan (Jordan) in 2001 and 2003 respectively. He is currently an Associate Professor in the Computer Science Department at the University of Jordan. His research interests include Image Processing, Pattern Recognition, Paper Watermarks, Document Analysis and Recognition, Multimedia applications, and Data Hiding.