The Minimum Number of Iterations Required to Replicate Binary Images using Discrete One-dimensional Dynamical Systems and a Retrocausal Updation Rule

Sugata Mitra Tataha Kim Laboratory NIIT University NH8, Delhi-Jaipur Highway Neemrana, District Alwar Rajasthan 301705 India

ABSTRACT

Agents affected by their own future states in a onedimensional discrete dynamical system (1-DDS) can replicate two-dimensional images. A novel and practical set of two rules have been developed in this paper to calculate the number of iterations required for exact replication. It is argued that retrocausal updation used by 1-DDS can replicate any ndimensional digital object. It is shown that the way iterations reach a final image are different for randomly generated images and non-random images. This paper suggests a simple method for replicating and detecting non-randomness in images.

General Terms

Image processing, image replication, causality, meaning

Keywords

Image replication, retrocausality, toroidal spacetime, discrete dynamical systems, cellular automata, agents

1. INTRODUCTION

A one-dimensional cellular automaton (1-CA) is an array of cells, each in one of a finite number of states. At each time step, the state of each cell is updated using a deterministic rule based on the state of the neighboring cells and its own state. Such 1-CA's have been extensively studied over decades.

1-CA's can replicate binary images. This is particularly useful in image processing and encryption applications, as in textile and other motif design, genetics research, spread of infectious diseases, security, statistical physics, artificial life and other fields. Previous work in this area include:

- Wolfram [1] discussed self-similar patterns on 1-CA's.
- Willson [2–4] studied the generation of fractals and fractal dimensions with linear rules.
- Culik and Dube [5] proved that linear rules will always generate regular behavior on any initial configuration.
- Fredkin [6] discussed fractal replicator on twodimensional cellular automata (2D CA).
- Mitra and Kumar [7] discussed fractal replication on 1-CAs with look-ahead (cell's own future state) and that replication can happen at time step 2^k , where *k* is a non-negative integer.
- In [5, 7] and in Gravner and Griffeath [8], it was independently observed that replication occurs at time step $2^{\lceil (\log 2(m-1)) \rceil}$ (m > 1 is the size of the initial

configuration).

Finally, Inabathini and Lakshmanan [9] carried out a mathematical analysis and reported a method to calculate the value of k. by proving that an image of size *m* gets replicated exactly at step 2^k where *k* is the least positive integer satisfying the relation $m/r \le t = 2^k$. Here r is the size of the neighborhood and t is the time step. Many of the references to past work above is, indeed, reported from this paper.

However, finding the value of k, such that 2^k represents the time step at which replication happens, is still not clear from the work above. Even in the latest work [9], the value of k when the inequality $m/r \le t = 2^k$ is not satisfied, remains unclear. In the present paper, we have used an experimental approach that determines the value of 2^k for all values of m (the 'width') and h (the 'height') of an image.

For cellular automata diagrams, it is customary [8] to show the time axis as vertically pointing downwards, while the agents or cells are show as a horizontal line. However, in the work that follows, we have represented the agents as a vertical column pointing upwards from 1 to h, and the time axis as horizontally pointing from left to right. This was done for ease of viewing images.

The method of replication used in this paper uses agents that can look ahead into their future states. Images or bitmaps can be fractally replicated by *time manipulated one-dimensional cellular automata* [7]. Instead of modified cellular automata, a vertical array of agents forming a one-dimensional discrete dynamical system (1-DDS) were used. Each agent can be in one of two states, 1 and 0, represented by black and white. Each agent's current state is updated by the current states of its two neighbors, above and below, and by its own state in the future. Since its own future state affects its present state, the system is retrocausal. The 1-DDS can update over a fixed number of time steps, t=1 to w, and then return to t=1. Each such cycle is an iteration.

If a bitmap is placed in the future of this 1-DDS, the agents will react to the bits of the image and change in accordance with the updation rules provided. It is known that a modified (retrocausal) Wolfram rule 150 will replicate the image in 2^n iterations [7], where n is a positive integer.

In this paper we show that:

1. The relationship between the minimum number of iterations required to replicate a binary bitmap, I_{min} ,

and the 'size' of the bitmap is a step function of the width of the bitmap, but with exceptions if the number of agents in the 1-DDS is an integral power of 2. This corrects earlier inadequate explanations [7,8,9].

- 2. Iterations have a physical basis if the image being replicated is considered a surface in toroidal spacetime.
- 3. The progression of successive iterations towards replicating the final image shows differences for random images when compared with equivalent non-random images.

It is important to note that there are many references to replication of images by cellular automata, but, except for [7], none that uses retrocausality - the ability to use future states to alter present ones.

2. EXPERIMENTAL PROCESS

The replicator used consists of a retrocausal one-dimensional discrete dynamical system [10] made up of agents [11]. The system consists of a line of discrete agents arrayed in a specific geometry [12]. Each agent can be in one of k finite states, and each agent's state is updated on every time step according to a deterministic rule based on the values of the neighboring agents and the future state of the agent being updated. It is assumed that each agent has knowledge of its future state. In the computational experiments described below, we have worked with a 1-DDS that is, a single line of agents, placed vertically, each in one of two states – black=1 or white=0.

We have used the following two-state model having a neighborhood of radius 1:

Number of states: k =2

Number of neighbors above and below: r = 1

Retrocausal Update Rule (RUR):

Agent $_{t+1} = (Above_t + Agent_{t+1} + Below_t) \mod 2$

This rule implies that every agent's future state at time t+1 exists, and is the cause of the agent's state at time t.

This update rule is a modification of the usual:

Causal Update Rule (CUR):

Agent $_{t+1} = (Above_t + Agent_t + Below_t) \mod 2$

CUR is called Rule 150 according to Wolfram's numbering system and is quite well known [1].

Since our retrocausal update rule involves the future states of agents in a 1-DDS, we can populate that future with a binary bitmap. This could be an image. If we do so and iterate the 1-DDS over this image using RUR, we observe image replication as expected. If we had, instead, used CUR for updation, the image would have been erased by the advancing 1-DDS.

3. REPLICATION EFFICIENCY

A black-and-white (binary) image of Leonardo Da Vinci was replicated using a retrocausal 1-DDS as described above. The

results are shown in figure 1.



Figure 1. Image replicated by a retrocausal 1-DDS over 512 iterations

The image used has 480(width)x585(height) pixels, and contains approximately 25% black pixels. It was exactly replicated in 512 iterations. For each iteration, how closely the iterated image matched the original was calculated. A plot of the percent match (henceforth called replication efficiency) against iteration number is shown in figure 2.



Figure 2. How progressive iterations approach 100% replication of the original image

Figure 2 shows that the replication efficiency reaches 100% at the 512th iteration. Iterated images get closer to matching with the original image, with peaks in replication efficiency every 2^n iterations. Exact match, i.e., 100% is achieved at n=9, the 512th iteration.

Each iteration, some shown in figure 1, involves application of update rule, RUR, until the end of the width of the picture, column 480 in this case, is reached. After this, the next iteration begins at column 1. It is important to realize that the column numbers are units of time and the width of this image represents time. Since our DDS is one-dimensional, the only spatial dimension it has is its height which is the number of agents in the system, 585 in this case. The vertical height axis is referred to as y, and the horizontal width axis as t.

If it is assumed that the t-axis is closed, that is, its value returns to zero after 480, a basis for why iterations restart from zero after reaching the maximum width of the image is established.

The update rule involves both neighbors of the agent being updated. This will cause a problem at both ends of the y-axis. This problem can be resolved by joining the ends of the yaxis. In other words, the y-axis is closed, and its value returns to zero after 585. This provides a consistent basis for applying RUR along the y-axis.

The shape of a spacetime formed by a closed spatial axis, y, and a closed temporal axis, t, is a two-dimensional, orthogonal torus as shown in figure 3. The torus shape removes problems related to edges – whether it is the maximum and minimum values on the y- axis, or the future and past on the t-axis. The application of such toroidal spacetime to cellular automata is not common in the literature, however, there have been some attempts [13].



Figure 3. Toroidal spacetime with one temporal and one spatial axis

An image or a group of pixels (bitmap) wrapped such that it fits exactly onto a torus such as the one shown in figure 3, would enable update rule RUR to apply over the time axis in the direction of the future until it covers the circumference of the time axis and returns to zero. This would complete a single iteration and the process would repeat, thereby replicating the image or bitmap repeatedly as shown in figures 1 and 2. We would have a physical analogue for the process of replication, provided we assume that any bitmap creates a toroidal spacetime around which it is draped.

It is also important to note that the bitmaps we are working with are two-dimensional, while the DDS we are using to replicate the bitmaps with are one-dimensional. Our 1-DDS is essentially a binary string, a column of agents, each in one of two states. The update rule RUR produces a new binary string at each time step. These strings when laid side by side produces a two-dimensional image. In other words, we are interpreting the time axis as an additional spatial axis to visualize an "image". What we are referring to as an "image" is a memory of past states of the column of agents, frozen as a spatial axis.

4. MINIMUM ITERATIONS NEEDED

The minimum number of iterations required for exact replication is called I_{min} , as multiples of I_{min} iterations will also produce exact replications. We know from the experiments above that $I_{min} = 2^n$, where n is a positive integer. In the earlier paper [7], it was suggested that I_{min} is related to the area of the image, and two special cases were discussed. However, no general method was suggested for calculating I_{min} for an image of width (w) and height (h). In this paper, we will derive such a method.

We need to know if I_{min} is related to the height of the image on the y-axis, the width of the image on the t-axis, or both. To do this, we will need to compare the I_{min} values for equivalent images of different heights and widths. We generated images consisting of random black and white dots to conduct these experiments.

First, it was checked whether an image consisting of random pixels would be equivalent to a non-random image with the same number of pixels. The image of Leonardo in figure 1 consists of 480x585 pixels (black or white dots) with 25% black pixels. A random bitmap was created by distributing the same number of black pixels randomly over an image of the same dimensions. This was replicated using the RUR on a 1-DDS. Figure 4 shows the replication efficiency between iterations 400-512.



Figure 4. Comparison of replication efficiencies for random and non-random images

The match percent data for the Leonardo image of Figure 1 showed a 76% correlation with that for the equivalent random bitmap used for Figure 4. More importantly perhaps, the peaks in both graphs are at identical powers of 2. We concluded that while the matching progresses differently for the two images, the number of iterations required for exact replication are identical. This was noticed for several images compared with equivalent random bitmaps. This difference between random and non-random images seems to indicate that height, width, and pixel density are not the only determinants that shape the replication efficiency graph. The distribution of pixels, whether random or non-random, affects the shape of the graphs in figure 4, even though they don't affect the value of I_{min} .

Hence, for the purpose of determining the relationship between I_{min} and the height and width of an image, random images were used for the experiments below.

Experiment 1: The y-axis height(h) of the image was varied

from 1 to 256 in steps of 1, while keeping the width (w) constant at 25. At each step, an equivalent random image was generated and iterated using RUR until exact replication was achieved. The results are shown in figure 5 up to h=69.



Figure 5. Iterations needed to reach 100% replication for images with varying heights and fixed width

Figure 5 shows that an image of width t=25 units will require $[log_2t + 1]$ =32 iterations to replicate irrespective of its height, except when h is an integral power of 2 and is $[log_2t + 1]$ =32 or less. In our example, h meets these conditions for values 2,4,8,16, and 32. At these exact values, I_{min}=h/2. Several other experiments confirmed these findings. The dependence of I_{min} is on the range of powers of 2 that the height and width of an image are in.

Experiment 2: The t-axis width(t) of the image was varied from 1 to 256 in steps of 1, while keeping the height (h) constant at 25. At each step an equivalent random image was generated and iterated using RUR until exact replication was achieved. The results are shown in figure 6 upto width=67.



Figure 6. Iterations needed to reach 100% replication for images with varying widths and fixed height

Figure 6 shows that for images with widths less than height, an image of fixed height (25 units in this case) will require 2^n iterations for an exact match, (where n is a positive integer) such that $2^{n-1} \le w \le 2^n$.

We can also write this as $n = \lceil log_2 t + 1 \rceil$. For example in Figure 6, for values of w<h, $I_{min}=2^4$ for widths where $2^3 <= w < 2^4$.

In general, $I_{min}=2^{\lceil log_2t+1 \rceil}$, except when h is a positive integral power of 2, as seen from experiment 1.

I_{min} is a step function of width, except when the height is an integral power of 2. To check this, heights of 63,64,65 and 128 were tested. Figure 7 shows the results. At h=63 where $2^5 < h <= 2^6$ and at h=65 where $2^6 < h <= 2^7$, I_{min} changes as a step function of width. However, at h=64, I_{min} becomes independent of width and remains constant at 32 which is 2^5 . Similarly, at h=128, I_{min} becomes independent of width and remains constant at 64 which is 2^6 . The individual graphs for each height is shown on the top of Figure 7, followed by the combined graph.



Figure 7. Relationship of Imin with width for different heights

Next, widths that were 63, 64 and 65 were tested. These are shown in Figure 8. As expected, at widths of 63 and 64 where $2^5 <= w < 2^6$, I_{min} is 2^6 . At a width of 65, where $2^6 <= w < 2^7$, I_{min} jumps to 2^7 . Also, as expected, this process is disrupted when the height is an integral power of 2, if its value does not exceed $2^{\lceil \log_2 t + 1 \rceil}$.



Figure 8. Relationship of Imin with height for different widths

We can generalize these results into the following rules:

For a bitmap of dimensions w (width), h (height) and a positive integer n:

1. $I_{min} = 2^{\lceil log_2w+1 \rceil}$, for all values of w when $h \neq 2^n$, or when $h => 2^{\lceil log_2w+1 \rceil}$

2. $I_{min} = 2^{\lceil log_2h-1 \rceil}$, for all values of h=2ⁿ when h<=2^{\lceil log_2w+1 \rceil}

 I_{min} is always determined by the integral powers of 2 on the time axis (w), except when the number of agents making up the 1-DDS, the y axis (h) value, is a positive integral power of 2 and is less than the upper end of the integral powers of 2 that w lies in. Hence, I_{min} can be made independent of w by adjusting the dimensions of a bitmap so that its h value (that is, the number of agents in the 1-DDS) is a positive integral power of 2, less than $2^{\lceil log_2 t+1 \rceil}$.

We can use these rules to predict the value of $I_{\mbox{\scriptsize min}}$ for a bitmap of any dimensions.

5. EXPERIMENTAL VERIFICATION

We used the rules above to predict the I_{min} values for bitmaps of different dimensions. In all cases the predictions matched the number of iterations required to replicate the bitmaps. To illustrate this, three examples of image replication are reported below.

Example 1: A bitmap image of an elephant of dimensions w=272, h=177 was replicated.



Figure 9. Replication of an elephant image

Since h is not an integral power of 2, Rule 1 above will apply and this predicts an I_{min} of 512, which matched the experimental result as shown in Figure 9.

Example 2: A bitmap image of a person of dimensions w=156, h=256 was replicated.



Figure 10. Replication of a person image

Since h in integer power of 2, Rule 2 above will apply and predicts an I_{min} of 128 which matched the experimental result as shown in Figure 10.

Example 3: A bitmap image of a butterfly of dimensions w=275, h=256 was replicated.



Figure 11. Replication of a butterfly image

Since h in integer power of 2, Rule 2 above will apply and predicts an I_{min} of 128 which matched the experimental result as shown in Figure 11.

6. DISCUSSION

6.1 Replication of n-dimensional objects

In the work described above, two-dimensional bitmap images were replicated. Such replication is possible for any digitally represented object because all such objects of any number of dimensions can be represented as a one-dimensional binary string [14]. Any such binary string can be 'folded' into a rectangle of sides h and w and replicated in 2ⁿ iterations by 1-DDS using the retrocausal update rule RUR in a toroidal spacetime. The replicated bitmap can then be 'unfolded' into the original binary string and finally, converted to the original digital object.

Example: We converted a word-processed document in the .docx format into a binary string folded into a rectangular bitmap of dimensions w=256 and h=371 (figure 12).



Figure 12. A" .docx" file converted into a binary bitmap image

This image was replicated in 256 iterations using the RUR as predicted by Rule 1 above. When 'unfolded' and renamed, the original document was recovered.

6.2 Effect of iteration on meaning

Random images and meaningful non-random images show differences in replication graphs as seen in figure 4. RUR

based Iterations seem to either destroy or create meaning. An example is shown in figure 13.



Image B after 1 iteration

Figure 13. The effect of a single iteration on two, nearly identical, images

Images A and B in figure 13 are different by about 17%, and nearly identical visually. Image A is the original image while image B is the result of 511 iterations, using RUR, on image A. A single iteration destroys all meaning in image A while a single iteration on image B produces an exact replication of the original image. Each pixel in these images are states of agents in a 1-DDS as described before. Figure 13 shows that the states of a few agents in a 1-DDS can affect the results of an iteration to exceptionally large extents.

The progression of iterations over time are irreversible. If an agent has a value of 1 at time t, it is not possible to determine what its value at time t-1 was. This is because the updation rule is such that the same state can be produced by different combinations of the states of itself and its upper and lower neighbors in the previous time step. Hence, RUR is deterministic in the forward, past to present to future, direction of time, but not the other way about.

Further, it is not possible to derive the image from its replication efficiency graph. For example, the difference between the replication efficiency graphs for an image being studied and an equivalent random image may suggest that the image being studied is a non-random image, but, in the absence of the original image, the replication efficiency graph alone cannot tell us what that image was. Further research may uncover a method to retrieve an image from a replication efficiency graph.

The process of replication of non-random images appears to be one where iterations create and destroy meaning in cycles, until the final image is reached.

Meaning in binary strings is not understood clearly although

some attempts have been made (e.g., [15]).

6.3 Applicability to other cyclical systems

The worldwide COVID-19 virus pandemic [16] shows cyclical patters that can be compared with 1-DDS iteration efficiency graphs. The growth of infections shows peaks and troughs over time, heading upwards. This is typical of the behavior of replication efficiency in bitmap replication by retrocausal agents in a 1-DDS. Figure 14 shows such a comparison.



Figure 14. COVID-19 infections worldwide compared to a replication graph of 1-DDS

Correlation does not imply causation and it is highly unlikely that the same mechanism is at work in both cases, however, that possibility cannot be ruled out. We found a 75% correlation between the graphs in Figure 14. However, if we construct an equivalent random bitmap for the image in Figure 14, the correlation between the Covid data and this equivalent random bitmap is found to be only 20%. This is shown in Figure 15.



Figure 15. Worldwide Covid cases between 31st May and 16th June 2020, compared with replication efficiency of an image and its equivalent random bitmap

Virus infection numbers involve neighborhood interactions to determine future states. If the mechanism of the spread of the Covid virus is like that of a 1-DDS replicating an image, Figure 15 tells us that image is non-random. If this is so, the I_{min} values for replication can be made constant and independent of time by changing the number of agents involved, as shown in section 4.

While unlikely and speculative, we could imagine that the graph of infections in figures 14 and 15 suggest a retrocausal 1-DDS iterating over time to replicate an *image*, or a pattern.

7. CONCLUSIONS

We conclude that any digital object can be replicated by onedimensional discrete dynamical systems using a retrocausal iterative rule where its own future determines its present state. The minimum number of iterations required to replicate an object using this method can be calculated by applying two simple rules.

One consequence of this work shows that the minimum number of iterations required to replicate a digital object (converted into a one-dimensional binary string) using a 1-DDS can be controlled and made independent of the dimension of the object by suitably altering the number of agents that constitute the 1-DDS.

Finally, we find that the replication efficiency graphs of random and non-random images as iterations proceed towards exact replication, are significantly different from each other, thereby offering a means, perhaps, for distinguishing between random and non-random ('meaningful') images.

8. REFERENCES

- Wolfram, S. (1983). "Statistical Mechanics of Cellular Automata," Reviews of Modern Physics, 55 601–644. http://lattice.ifsc.usp.br/~lattice/oldlattice/artigowolfram-cellular-autom.pdf
- [2] Willson, S. J. "Cellular Automata Can Generate Fractals," *Discrete Applied Mathematics*, 8(1), 1984 pp. 91–99.doi:10.1016/0166-218X(84)90082-9.
- [3] Willson, S. J. "Computing Fractal Dimensions for Additive Cellular Automata," *Physica D: Nonlinear Phenomena*, **24**(1–3), 1987 pp. 190–206. doi:10.1016/0167-2789(87)90074-1.
- Willson, S. J. "Growth Rates and Fractional Dimensions in Cellular Automata," *Physica D: Nonlinear Phenomena*, **10**(1–2), 1984 pp. 69–74. doi:10.1016/0167-2789(84)90250-1.
- [5] Culik II, K. and Dube, S. "Fractal and Recurrent Behavior of Cellular Automata," *Complex Systems*, 3(3), 1989 pp. 253–267. complex-systems.com/pdf/03-3-3.pdf.
- [6] Fredkin, E. "An Informational Process Based on Reversible Universal Cellular Automata," *Physica D: Nonlinear Phenomena*, **45**(1–3), 1990 pp. 254–270. doi:10.1016/0167-2789(90)90186-S.
- [7] Mitra, S. and Kumar, S. "Fractal Replication in Time-Manipulated One-Dimensional Cellular Automata," *Complex Systems*, 16(3), 2006 pp. 191–197. complex-

systems.com/pdf/16-3-1.pdf.

- [8] Gravner, J. and Griffeath, D. "The One-Dimensional *Exactly 1* Cellular Automaton: Replication, Periodicity, and Chaos from Finite Seeds," *Journal of Statistical Physics*, **142**(1), 2011 pp. 168–200. doi:10.1007/s10955-010-0103-9.
- [9] Inabathini, S. & L. J. Replication of a Binary Image on a One-Dimensional Cellular Automaton with Linear Rules. *Complex Systems*. 2018;27(4):415–430. Available from: 10.25088/ComplexSystems.27.4.415
- [10] An introduction to discrete dynamical systems, Internet: "Math Insight" https://mathinsight.org/discrete_dynamical_system_intro duction
- [11] Bonabeau, Eric (2002). Agent-based modeling: Methods and techniques for simulating human systems, PNAS vol.
 99 suppl. 3, pp7280–7287. https://www.pnas.org/content/99/suppl_3/7280.
- [12] Shiffman, D. (2012). Chapter 7, The Nature of Code Simulating Natural Systems with Code. https://www.amazon.com/Nature-Code-Simulating-Natural-Processing/dp/0985930802
- [13] Rouhaud, J. (2000). Cellular automata and consumer behaviour, European Journal of Economic and Social Systems 14 N° 1 37-52. https://ejess.edpsciences.org/articles/ejess/pdf/2000/01/ro uhaud.pdf?access=ok
- [14] Duhamel, P. and Kieffer, M. Joint Source-Channel Decoding (2010), Academic Press, https://doi.org/10.1016/C2009-0-19065-7
- [15] Mitra, Sugata.(2002) 'Meaning in Binary Strings', Introduction to Multimedia Systems, Academic Press, pp. 151. Also downloadable from https://7551850c-18fa-45d6-98bc-8e2829cc3850.filesusr.com/ugd/369ee5_d5ddac04b3e14 a37939ed8b1f8f521f3.pdf
- [16] Corona Virus Data (2020), from ourworldindata.org https://ourworldindata.org/coronavirus-source-data