Dithering with Pascal Cellular Automaton

Yvan Richard[0000−0003−4497−3843]

govyvan@hotmail.com
http://www.linkedin.com/in/yvanrichard

Abstract. In this article, I present a cellular automaton (CA) which evolve according to a set of rules derived from a well known combinatorial structure, the Pascal's Triangle. With only 2 neighbour cells, this probabilistic cellular automaton (PCA) produces point sampling commonly observed in high-quality digital dithering and halftoning techniques.

Keywords: Halftoning and Dithering · Cellular Automata · Error Diffusion · Image Rendering and Quality · Pascal Triangle · Quantization · Discrete Sampling

1 Introduction and Motivation

To my surprise, I found that a simple Cellular Automata (CA) coupled with — Pascal's Triangle — as probabilistic rules, was able to produce high-quality halftone images. These images presents a point sampling signature that effectively simulates the pleasing texture and detail typically achieved with Frequency Modulation (FM) screening and specifically, Error Diffusion (ED) techniques. While cellular automata and error diffusion operate on different principles, they both share similarities in their spatial and temporal dependencies but also in their predictability, to some extent.

In cellular automata, each cell's state depends on its neighbours, showing spatial dependencies, and the system evolves in discrete steps according to predefined rules, indicating temporal dependencies. Similarly, in error diffusion, pixel quantization error spreads to neighbour pixels, showing spatial dependencies, and the process iterates over the image in a stepwise manner, with each pixel's quantization affecting subsequent ones, demonstrating temporal dependencies.

In general, cellular automata are governed by a set of predefined rules leading to a deterministic evolution. However, by introducing element of randomness or probability into their ruleset, probabilistic cellular automata (PCA) exhibit more complex dynamics and simulate stochastic processes with greater fidelity. Similarly, error diffusion algorithms are usually deterministic, but the addition of random noise to the quantization error or diffusion weights reduce the visibility of quantization artefacts, leading to more visually pleasing outputs.

However, beyond these spatial and temporal similarities, cellular automata and error diffusion differs in a fundamental aspect, the way data are quantified.

Typically, cellular automata computations are performed using binary or integer values, reflecting the discrete nature of the states. Whereas error diffusion utilizes fractional weights to spread the quantization error of pixel values across neighbour pixels, resulting in continuous floating-point data calculations.

By recognising these characteristics, we can effectively assess how this novel approach utilising Cellular Automata and Pascal's Triangle compares to the established effectiveness of Error-Diffusion technics in generating high-quality halftone images. The combination of these elements in the context of digital dithering and halftoning not only enhances our current understanding but also opens up further research opportunities in both digital imaging techniques and computational theory.

1.1 Error Diffusion

Error diffusion is a digital image processing technique, commonly employed in halftoning — a process converting continuous-tone images into binary images while approximating the original appearance. Widely embraced across industries such as printing, publishing, and display technologies, error diffusion methods are pivotal in generating high-quality dithered images. This technology finds application in devices like inkjet printers and e-ink displays.

Error-Diffusion can be described as follows: the algorithm scans the image, row by row, and pixel by pixel, the current pixel value plus a feedback error is compared to a threshold, the result of the comparison assigns an output value, e.g. 0 or 1 to the halftone image and the quantization error is distributed to neighbour pixels using predefined coefficients. By distributing the error across the image, the error diffusion process tends to smooth out abrupt transitions between different intensity levels, maintaining the overall tonal balance and detail in the final output. Below are the Floyd and Steinberg coefficients, the ($\square$) indicates the currently processed pixel and ($-$) indicates the previous pixel.

$$\frac{1}{16}\begin{bmatrix} - & \square & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

Over the past decades, researchers and specialists have explored numerous variations and extensions of error diffusion algorithms to address specific challenges, such as reducing artefacts, improving performance, and adapting to different types of input images and outputs, encompassing multilevel, color and tone dependencies. These variations include modifications to the diffusion coefficients, error weighting schemes, scan path, threshold modulation, noise shaping and incorporation of perceptual models, leading to a substantial body of literature, articles and patent filings, [1–12].

1.2 Pascal's Triangle

The Pascal's Triangle is a fascinating mathematical structure named after the French mathematician Blaise Pascal [13], although it was known to many mathematicians before him. It finds application across various fields due to its remarkable properties and connections to different areas of mathematics, ranging from combinatorics to algebra, from probability theory to geometry and extending into computer science and physics [14,15]. Donald E. Knuth offers us this amazing and insightful view of the rich mathematical landscape found in Pascal's Triangle:

> *There are so many relations present in Pascal's triangle that when someone finds a new identity, there aren't many people who get excited about it anymore, except the discoverer!*

It's a triangular array of numbers where each number is the sum of the two numbers directly above it. The triangle starts with a single 1 at the top, and each row thereafter is constructed by adding adjacent numbers in the previous row. Here's an example of the first few rows:

$$
\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1
\end{array}
$$

In image processing Pascal's triangle serves as a mathematical tool for analyzing, manipulating, and enhancing digital images [16–18]. One of its primary applications lies in filtering techniques, where it serves as a cornerstone for generating convolution kernels used in tasks such as image blurring and sharpening. By extracting Binomial coefficients $\binom{n}{k}$[1]. from Pascal's triangle rows, low-pass 2D binomial filters aka. Gaussian filter approximation can be efficiently constructed to smooth or blur images.

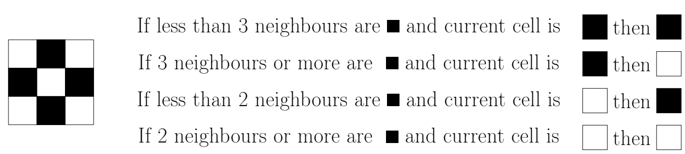
1	2	1
2	4	2
1	2	1

[1] $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the number of ways to select k out of n objects, when order does not matter

1.3 Cellular Automata

Cellular automata were first introduced and developed by mathematicians John von Neumann [19] but it was the work of John Conway [20] that brought this computational model into the spotlight with his creation of the Game of Life. Since then, numerous researchers have contributed to the advancement of cellular automata theory and applications over the years. Stephen Wolfram extensively studied and popularized them in his book "A New Kind of Science" [21].

Typically, they consist of a grid of cells, each of which in a finite number of states. The state of each cell evolves over discrete time steps based on a set of rules that determine how the state of a cell changes based on its current state and the states of its neighbour cells. The evolution of cells can occur in two different ways: *in synchronous updating*, all cells evolve simultaneously at each time step, while *in asynchronous updating*, cells evolve sequentially rather than simultaneously. Below the Von Neumann neighbourhood scheme of a 2D cellular automaton and it set of deterministic rules, each cell on the grid interacts with its immediate left, right, top, and bottom neighbours.



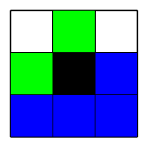
In image processing cellular automata have found various applications, including filtering and enhancement (such as edge enhancement, noise reduction, and image sharpening), segmentation (dividing an image into regions or objects of interest), texture synthesis, features extraction, compression, restoration, and reconstruction, among others [22,23] . They also intersect in fascinating ways in the realm of digital art, pattern generation, and procedural content creation [24]. In generative art installations, cellular automata create interactive artworks that respond to environmental inputs or user interactions [25].

Overall, cellular automata can exhibit remarkably complex behaviours, including the emergence of patterns, self-replication, and even universal computation. They have been used to model phenomena such as the spread of forest fires, traffic flow, the behaviour of fluids, the growth of biological organisms, spanning also into the enigmatic realm of quantum mechanics [26]. With the advancement of machine learning techniques, the concept of training cellular automata to achieve desired behaviours or outcomes has gained attention, such as developing adaptive, intelligent systems capable of self-learning and evolving to meet specific objectives [27–29].

2 Proposed Method

In it simplest form, I showcase a new dithering method, leveraging cellular automaton principles and Pascal's triangle as probabilistic rules to sample discrete values based on the values of two preceding neighbour pixels. The sampled value is then added to the current pixel value and thresholded, the resulting quantization error is updated for the current pixel and the process repeats iteratively.

In contrast to standard error diffusion, which typically reads one pixel ■ and updates the values of the next four neighbour pixels ■, our method reads the values of two preceding neighbour pixels ■ and updates the value of the current pixel ■ accordingly.



We can view this new processing scheme as a dual to the error diffusion algorithm since it shares most of its principles, except for its spatial and temporal dependencies and the fact that quantization error remains a discrete process.

Pascal's rows as rules Given an 8-bit continuous-tone grayscale image, where each pixel's value ranges from 0 (representing black) to 255 (representing white), we generate Pascal's Triangle up to 256 rows, corresponding to the available levels of grayscale intensity.

Then, the binomial coefficients $\binom{n}{k}$ for each row of Pascal's Triangle are summed and normalized[2] to create a cumulative distribution function (CDF). This function allows for the sampling of new discrete values between 0 and 255 by leveraging the fact that the n-th row of Pascal's Triangle (starting from 0) has n+1 entries. Below, the Pascal's triangle cumulative distribution function generation (left-aligned form), up to the 5-th row.

Pascal's Triangle Coefficients	Cumulative sum of Coefficients	Normalized Cumulative sum (CDF)
Row 0 : 1	1	1
Row 1 : 1 1	1 2	$\frac{1}{2}\ 1$
Row 2 : 1 2 1	1 3 4	$\frac{1}{4}\ \frac{3}{4}\ 1$
Row 3 : 1 3 3 1	1 4 7 8	$\frac{1}{8}\ \frac{4}{8}\ \frac{7}{8}\ 1$
Row 4 : 1 4 6 4 1	1 5 11 15 16	$\frac{1}{16}\ \frac{5}{16}\ \frac{11}{16}\ \frac{15}{16}\ 1$
Row 5 : 1 5 10 10 5 1	1 6 16 26 31 32	$\frac{1}{32}\ \frac{6}{32}\ \frac{16}{32}\ \frac{26}{32}\ \frac{31}{32}\ 1$

[2] The sum of the coefficients in the n-th row of Pascal's Triangle is 2^n.

Sampling Pascal's rules Given the values of the two preceding neighbour pixels, A and B, and the value of the current pixel, C, the discrete sampling process follows these steps:

1. Calcul the absolute difference, $\mathrm{Diff} = |A - B|$
2. Retrieve the corresponding row from Pascal's CDF, $\mathrm{PCDF}(\mathrm{Diff} + 1)$
3. Generate a random value from the uniform distribution, $\mathrm{Rand} = U(0, 1)$
4. Find the first $\mathrm{PCDF}(\mathrm{Diff} + 1)$ value that is greater than or equal to Rand.
5. Retrieve the discrete Index of this PCDF value
6. Calculate the new value for C, New $=$ Index $+$ min(A,B)

$$\text{In summary: New} = \begin{cases} A & \text{if } A = B \\ [A, B] & \text{otherwise} \end{cases}$$

Dithering with Pascal's rules Finally, the current pixel value plus the New value is compared to a threshold (e.g. 255), the result of the comparison assigns an output value, e.g. 0 or 1 to the halftone image and the current pixel value, C, is updated with the resulting quantization error. Et voilà! The Python implementation can be found at https://github.com/GovYvanR/Dithering-with-Pascal-Triangle, this version already implement the possibility of modulating the threshold.

Algorithm 1 Dithering with Pascal CA

```
procedure DitherImage(input_image)
    Initialize an empty dithered image
    Initialize PascalCDF
    for each pixel (x, y) in the input image do                ▷ Serpentine Scan
        pixel_A ← input_image(x, y − 1)
        pixel_B ← input_image(x − dir, y)
        pixel_C ← input_image(x, y)
        discrete_Value ← SamplePascalCDF(pixel_A, pixel_B, PascalCDF)
        new_pixel_C ← pixel_C + discrete_value
        dithered_image(x, y) ← new_pixel_C
        quant_error ← new_pixel_C − 255
        input_image(x, y) = quant_error                ▷ Update current pixel value
    end for
    return dithered image
end procedure
procedure SamplePascalCDF(A, B, PCDF)
    diff_AB ← abs(A - B)
    Rand ← Random number between 0 and 1                ▷ Uniform distribution
    Index ← PCDF(diff_AB, Rand)
    return Index + min(A,B)
end procedure
```

3 Results

To evaluate the effectiveness of this new dithering method, in its simplest form, we conduct a comparative analysis against established, simple, and efficient error diffusion techniques. All of these techniques will be processed using a serpentine scan of the image.

Fig. 1: **Top row**, left: Floyd-Steinberg ED [1] (4 neighbour pixels - deterministic); right: Victor Ostromoukhov tone dependent ED [9] (3 neighbour pixels - deterministic). **Bottom row**, left: Robert Ulichney ED [3] - FS with 50% random weights (4 neighbour pixels - non-deterministic); right: our new method (2 neighbour pixels - non-deterministic).

Finally, given the non-deterministic nature of our solution, we conduct the following experiment: we generate 255 iterations of our algorithm on the same image and sum these iterations to reconstruct an 8-bit image. Subsequently, we compare the reconstructed image with the result obtained by performing Floyd-Steinberg dithering with 50% random weights. And we calculate the Structural Similarity Metric Index (SSMI) [30] to evaluate the quality of the reconstruction.

Fig. 2: left: Robert Ulichney ED [3] - FS with 50% random weights (4 neighbour pixels - non-deterministic); right: our new method (2 neighbour pixels - non-deterministic).

	FS with 50% random weights	Our method
SSMI Index	0.8003	0.7632

4 Conclusions and Future Work

In this article, I presented a new method for generating high-quality dithered images based on the cellular automata principle and Pascal's Triangle as probabilistic rules. Dithering and halftoning are well-established techniques, and this new method offers valuable insights into potential enhancements in this field. In its simplest form, the method addresses challenges in simulating detailed textures and gradients for image processing. This new framework and its discrete singularity suggest future improvements, such as extending the neighbour scheme from multinomial coefficients or Hilbert space-filling curves for image raster scans, color and multilevel, plane and tone dependent, threshold modulation, output-dependent feedback, noise shaping analysis, and optimization based on higher-order Markov models, as well as exploring 3D dithering techniques. These enhancements provide opportunities for more sophisticated image processing techniques. However, the significance of this method extends beyond its immediate applications.

I've called it the "*The Blue Tip of the Iceberg*" as this article only presents a small and visible portion of a much larger and complex framework. The majority of its mass lies beneath the surface of the water, hidden from view.

Acknowledgments. The author would like to express gratitude to numerous individuals who were involved at different degree in the present paper, particularly those who continue to pursue their dreams.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

References

1. Floyd, R.W., Steinberg, L.: An Adaptive Algorithm for Spatial Greyscale. Proceedings of the Society for Information Display **17**(2), 75–77 (1976)
2. Jarvis, J., Judice, C., Ninke, W.: A survey of techniques for the display of continuous tone pictures on bilevel displays. CGIP **5**(1), 13–40 (March 1976)
3. Ulichney, R.: Digital halftoning. MIT Press, Cambridge, MA, USA (1987)
4. Eschbach, R., Knox, K.T.: Error-diffusion algorithm with edge enhancement. J. Opt. Soc. Am. A **8**(12), 1844–1850 (Dec 1991). https://doi.org/10.1364/JOSAA.8.001844, http://josaa.osa.org/abstract.cfm?URI=josaa-8-12-1844
5. Velho, L., Gomes, J.d.M.: Digital halftoning with space filling curves. In: Proceedings of the 18th annual conference on Computer graphics and interactive techniques. pp. 81–90. SIGGRAPH '91, ACM, New York, NY, USA (1991). https://doi.org/http://doi.acm.org/10.1145/122718.122727, http://doi.acm.org/10.1145/122718.122727
6. Eschbach, R.: Reduction of artifacts in error diffusion by means of input-dependent weights. Journal of Electronic Imaging **2**(4), 352–358 (1993). https://doi.org/10.1117/12.151415, http://link.aip.org/link/?JEI/2/352/1
7. Shiau, J.N., Fan, Z.: Set of easily implementable coefficients in error diffusion with reduced worm artifacts. In: Electronic imaging (1996), https://api.semanticscholar.org/CorpusID:2833338
8. Knox, K.T.: Evolution of error diffusion pp. 448–458 (1998). https://doi.org/10.1117/12.334588, +http://dx.doi.org/10.1117/12.334588
9. Ostromoukhov, V.: A simple and efficient error-diffusion algorithm. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 567–572. SIGGRAPH '01, ACM, New York, NY, USA (2001). https://doi.org/http://doi.acm.org/10.1145/383259.383326, http://doi.acm.org/10.1145/383259.383326
10. Li, P., Allebach, J.P.: Tone-dependent error diffusion. IEEE Transactions on Image Processing **13**(2), 201–215 (2004)

11. Levien, R.: Output dependant feedback in error diffusion halftoning. IS&T Imaging Science and Technology **1**, 115?118 (1992)
12. Lau, D.L., Arce, G.R., Gallagher, N.C.: Green noise digital halftoning. Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269) **2**, 39–43 vol.2 (1998), https://api.semanticscholar.org/CorpusID:17703328
13. Pascal, B.: Traité du Triangle arithmétique, avec quelques autres petits traités sur la même matière, Par Monsieur Pascal. Guillaume Desprez (1665), https://books.google.es/books?id=UqgUAAAAQAAJ
14. Bogomolny, A.: Pascal triangle properties (2018), https://www.cut-the-knot.org/arithmetic/combinatorics/PascalTriangleProperties.shtml
15. Farina, A., Giompapa, S., Graziano, A., Liburdi, A., Ravanelli, M., Zirilli, F.: Tartaglia-pascal's triangle: a historical perspective with applications. Signal, Image and Video Processing **7**(1), 173–188 (2013)
16. Gonzalez, R.C., Woods, R.E.: Digital image processing. Prentice Hall, Upper Saddle River, N.J. (2008), http://www.worldcat.org/search?qt=worldcat_org_all&q=013505267X
17. Boutin, M.: The pascal triangle of a discrete image: Definition, properties and application to shape analysis. Symmetry, Integrability and Geometry: Methods and Applications (Apr 2013). https://doi.org/10.3842/sigma.2013.031, http://dx.doi.org/10.3842/SIGMA.2013.031
18. Mansour, E.H.A., Bretaudeau, F.: A novel edge detection method based on efficient gaussian binomial filter. International Journal of Advances in Intelligent Informatics **7**(2), 211–224 (2021). https://doi.org/10.26555/ijain.v7i2.651, https://ijain.org/index.php/IJAIN/article/view/651
19. von Neumann, J.: Theory of Self-Reproducing Automata. University of Illonois Press, Champain, IL (1966)
20. Games, M.: The fantastic combinations of john conway's new solitaire game "life" by martin gardner. Scientific American **223**, 120–123 (1970)
21. Wolfram, S.: A New Kind of Science. Wolfram Media (2002), https://www.wolframscience.com
22. Afandi, A.J.: Cellular automata-based image processing: A review. International Journal of Computer Applications **82**(15), 6–12 (2013)
23. Mittal, S., Kapoor, R.: Cellular automata and its applications in image processing. International Journal of Advanced Research in Computer Science and Software Engineering **5**(10), 1283–1287 (2015)
24. Gumin, M.: Wavefunctioncollapse is constraint solving in the wild. arXiv preprint arXiv:1610.09282 (2016)
25. Milesi, L.: Generative art installations: A computational ecosystem for design creativity. International Journal of Arts and Technology **5**(1), 48–66 (2012)
26. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum cellular automata: The physics of quantum information and computation. Physical Review Letters **90**(16), 160402 (2003)
27. Doe, J.: Training cellular automata for image processing tasks using evolutionary algorithms. IEEE Transactions on Evolutionary Computation (2020)
28. Smith, J.: Reinforcement learning for cellular automata rule optimization. Journal of Artificial Intelligence Research (2019)
29. Johnson, A.: Neural cellular automata: Learning rules for complex systems. In: Neural Information Processing Systems (2021)
30. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861