



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

TRANSLATION AUTOMATA APPLIED IN VISUAL ARTS

PŘEKLADOVÉ AUTOMATY APLIKOVANÉ V UMĚNÍ

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR

AUTOR PRÁCE

DZIYANA KHRYSTSIUK

SUPERVISOR

VEDOUCÍ PRÁCE

prof. RNDr. ALEXANDR MEDUNA, CSc.

BRNO 2024

Bachelor's Thesis Assignment



153864

Institut: Department of Information Systems (DIFS)
Student: **Khrystsiuk Dzyana**
Programme: Information Technology
Title: **Translation Automata Applied in Visual Arts**
Category: Theoretical Computer Science
Academic year: 2023/24

Assignment:

1. Based upon the supervisor's instructions, study selected types of automata, such as cellular or two-dimensional automata, which are suitable for this bachelor thesis.
2. Introduce modified versions of these automata. Consult the supervisor about this modification.
3. Study the properties of these automata and the languages accepted by them. Follow the supervisor's suggestions concerning this study.
4. Based on the supervisor's instructions, select suitable fields of visual art. Based upon the automata introduced in 2, design several applications in the selected fields.
5. Implement the applications designed in 4.
6. Evaluate the achieved results of this bachelor thesis. Discuss its possible further development.

Literature:

- Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Volume 1-3, Springer, 1997, ISBN 3-540-60649-1
- Brinkmann, R.: *The Art and Science of Digital Compositing: Techniques for Visual Effects, Animation and Motion Graphics*, 2nd edition, Morgan Kaufmann, 2008, ISBN 978-0-123-70638-6
- Gažo, M.: *Vícédimensionální automaty a jejich aplikace v umění*. Brno, 2021. Bachelor's Thesis. Brno University of Technology, Faculty of Information Technology. 2021-06-14. Supervised by Meduna Alexander. Available from: <https://www.fit.vut.cz/study/thesis/23696/>

Requirements for the semestral defence:
Points 1 to 3.

Detailed formal requirements can be found at <https://www.fit.vut.cz/study/theses/>

Supervisor: **Meduna Alexandr, prof. RNDr., CSc.**
Head of Department: Kolář Dušan, doc. Dr. Ing.
Beginning of work: 1.11.2023
Submission deadline: 9.5.2024
Approval date: 30.10.2023

Abstract

The goal of this work is to create a tool based on a translation automaton that supports artists by offering a structured approach to visual art. As a result of this work, a web based application, now accessible online, was developed with the use of React framework. The tool implements translation rules using cellular automaton, providing artists with variety of creative transformations that enable them to envision potential changes in their existing work and discover new possibilities for post-production image editing.

Abstrakt

Cílem této práce je vytvořit nástroj založený na překladovém automatu, který podporuje umělce poskytnutím strukturovaného přístupu k výtvarnému umění. Výsledkem této práce je webová aplikace dostupná online, která byla vyvinuta s využitím frameworku React. Nástroj implementuje překladové pravidla pomocí celulárního automatu a poskytuje umělcům různé kreativní transformace, které jim umožňují představit si potenciální změny jejich stávajících děl a objevit nové možnosti postprodukčních úprav obrazů.

Keywords

cellular automaton, translation automaton, abstract art, image editing, artistic application

Klíčová slova

celulární automat, translační automat, abstraktní umění, editace obrázků, umělecká aplikace

Reference

KHRYSTSIUK, Dziyana. *Translation Automata Applied in Visual Arts*. Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor prof. RNDr. Alexandr Meduna, CSc.

Rozšířený abstrakt

Vizuální umění je výkonný komunikační prostředek, který překračuje hranice běžného jazyka a zprostředkovává složité příběhy a emoce. Umělci využívají prvky jako barvu, tvar a texturu k vytváření poutavých zážitků. Formalizace struktury umění jako jazyka umožňuje jeho reprezentaci prostřednictvím konečného stavového automatu, což nabízí nový způsob interpretace a zpracování vizuálního umění.

Hlavním cílem této práce je vytvořit nástroj založený na překládovém automatu, který poslouží umělcům při tvorbě. Nástroj je inspirován abstraktními a geometrickými olejomalbami z počátku 20. století a kombinuje tyto umělecké styly s moderní estetikou a generativními algoritmickými uměleckými technikami. Takový nástroj může sloužit jako postprodukční pomůcka nebo zdroj inspirace, pomáhá umělcům vizualizovat možné transformace a otevírá nové cesty v jejich tvůrčím procesu.

Při zkoumání moderního umění se studium zaměřuje na abstraktní a geometrické styly, jako je orfismus, suprematismus a neoplasticismus, snaží se pochopit a zachovat jejich významné techniky a charakteristiky v rámci aplikace. Přehled teoretických základů překládových a buněčných automatů poskytuje podklad pro implementaci systému. Následný vývoj a architektura aplikace, včetně komplikací, které se vyskytly a byly vyřešeny, jsou podrobně popsány v rámci celé práce. Práci uzavírá ucelený přehled všech integrovaných pravidel a několik příkladů použití aplikace, a také některé návrhy na budoucí vylepšení.

Propojením teoretických poznatků s praktickou aplikací vzniká systém, který podněcuje uživatele k tvůrčímu experimentování s transformací obrazu. Architektura postavená na frameworku React podporuje zpracování a úpravy obrázků prostřednictvím různých vlastních komponent. Kromě manipulace s obrazem a unikátních vizuálních transformací řízených pravidly celulárního automatu obsahuje aplikace také kvantizaci barev, automatické generování pravidel a metriky umělecké kvality.

Translation Automata Applied in Visual Arts

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of prof. RNDr. Alexandr Meduna, CSc. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Dziyana Khrystsiuk
May 8, 2024

Acknowledgements

I am pleased to express my greatest appreciation to my supervisor, Prof. RNDr. Alexandr Meduna, CSc., for his valuable advice, motivation, and continuous support. I also would like to mention a great teacher and visionary, Mgr. Tomáš Staudek, Ph.D., for that his course on Computer Art and his personal enthusiasm deeply influenced my choice of theme and my general overview.

Contents

1	Introduction	3
2	Brief Review of Modern Art Aesthetics	4
3	Two-Dimensional Automata	14
4	Architecture and Features	22
5	Experimenting with the Rule Set	26
6	Conclusion	39
	Bibliography	40
A	Contents of the included storage media	43

List of Figures

2.1	An 1855 Punch cartoon satire about daguerreotype photography	5
2.2	Works of Wassily Kandinsky	9
2.3	Works of Paul Klee	10
2.4	Works of Piet Mondrian	11
2.5	Works of Robert Delaunay	12
2.6	Works of Kazimir Malevich	13
3.1	Neighborhood models of Von Neumann (left) and Moore (right)	17
3.2	Margolus neighborhood, also known as block cellular automaton	17
3.3	Periodic boundary graphic representation in 2D	18
3.4	Example of the Wolfram's Four Classes of Behavior	18
3.5	Static examples of selected totalistic rules.	19
3.6	Visualisation of Single Rotation rule with frame interpolation	20
5.1	Color rule in practice	26
5.2	Automated generation of recoloring rules from image palette	27
5.3	Move rule in practice	28
5.4	Several move rules with different directions applied in sequence	28
5.5	Zoom rule in practice	29
5.6	Zooming artefact on non-square objects in an asymmetrical composition . .	30
5.7	Cyclic rule in practice	30
5.8	Example of sequential change	31
5.9	Box Blur rule in practice	32
5.10	Median Mold rule in practice. Quantified original [16] transformed with 16 iterations	33
5.11	Visual effects of a brushstroke	33
5.12	Lines dissolution effect of Median Mold rule	34
5.13	Neighborhood XOR rule in practice	35
5.14	Difference between even and odd numbers of neighbors for XOR rule	35
5.15	Effect of a XOR rule similar to edge detection algorithm	36
5.16	Different rules applied sequentially	36
5.17	Improvisation on Mondrian [31]	37
5.18	Improvisation on Kandinsky [18]	37
5.19	Improvisation on Malevich [27], diptych	38
5.20	Improvisation on Klee [20]	38
5.21	Improvisation on Delaunay [8], diptych	38

Chapter 1

Introduction

Visual art serves as a sophisticated form of communication that surpasses the conventional boundaries of language to convey intricate ideas and narratives. Artists leverage imagery, harnessing elements such as color, form, and texture to create experiences that engage the viewer's senses and emotions. By formalizing the structure of art, we establish it as a formal language. Representing this formal language can be achieved through the application of a finite state automaton, which provides a structured approach to interpreting and processing the elements of visual art.

The main goal of this work is to create a tool suitable for artistic purposes, that is based on translation automaton. The application is designed with abstract and geometric oil paintings from the early twentieth century in mind and aims to integrate them with the precise aesthetics and generative algorithmic art techniques available today. Such tool can serve as a post-production resource or as a source of inspiration, enabling artists to visualize potential future transformations and discover previously unexplored possibilities in their creative process.

Chapter 2 begins with a brief study of the key features of modern art, focusing specifically on abstract and geometric styles such as orphism, suprematism, and neoplasticism. This chapter examines the significant techniques and characteristics of these art movements, aiming to understand, preserve, and celebrate them in the application. In Chapter 3, the theoretical foundations of translation automaton and cellular automaton are explored, providing a solid basis for subsequent implementation. The next chapter includes explanation of the application's architecture and the framework used in its development. It also discusses additional features, the complexities encountered and resolved during implementation, and other technical details (Chapter 4). The work culminates in Chapter 5, which offers an overview of all the integrated rules and a demonstration of the application's usage and capabilities. Finally, in Chapter 6, the achievements are summarized, and suggestions for future improvements to the developed application are proposed.

The latest version of the application developed in scope of this work is available online on the following link:

https://dionysusbacchus.github.io/Art_CA/

Chapter 2

Brief Review of Modern Art Aesthetics

To develop a useful artistic tool, one must understand various artistic methods. Though numerous art styles and movements exist, the focus will be on a select few. In this chapter, examining art movements from the avant-garde period — recognized for their streamlined shapes, abstract themes, and vibrant use of color — provides insight into their influence and applications in contemporary artistic practice.

Premise in Technological Change

Historically, painting has been closely linked to realistic themes and objects, with the primary genres being portraiture, landscape, and still life. Art served as a representation and depiction of the world around the painter. Capturing real-life objects accurately is a complex task, requiring understanding of the proper projection of the three-dimensional world onto two dimensions of a canvas. Achieving precise matches in color, shades, and forms demands significant skill, knowledge, and effort. Despite these challenges, traditional painting was one of the few methods available to represent subjects and people. This made skilled painters essential for tasks such as creating family portraits or depicting biblical scenes in church interiors, making realistic painting an important and relatively profitable profession. This persisted until the advent of photography.

Exploration in photography began as early as the 18th century but did not gain significant success until the mid-19th century [38]. Initially, it was a pursuit for skilled chemists rather than the general public. Even after the development of daguerreotype photography, which brought to market family photos as an alternative to painted portraits, the process involved long exposures and inconsistent quality, illustrated in Figure 2.1. The turning point came in 1884 when George Eastman introduced photographic film as an alternative to photographic plates. His Kodak camera, released in 1888, brought amateur photography to life.

With this advancement, anyone could capture a realistic image of a beautiful tree, their child, or even create a self-portrait quickly and easily with a ready to use simple box. No longer were painting skills, oils, or knowledge of anatomy and color theory required. This new creative tool fulfilled the representational purpose of imagery, effectively undermining the main aspect of oil painting and causing a major upheaval in visual art. Artists worldwide had to reconsider their motivations and question traditional methods and constraints, leading to the emergence of modern art movements.

PUNCH'S ALMANACK FOR 1855.

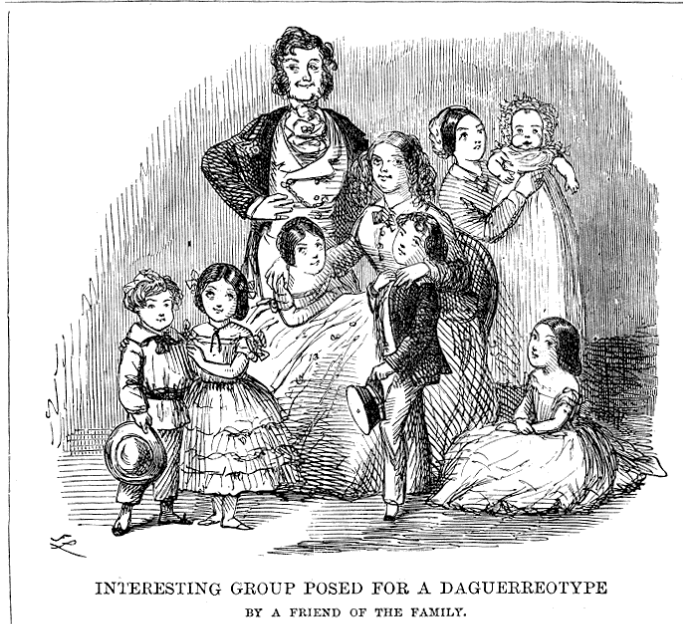


Figure 2.1: An 1855 Punch cartoon satire about daguerreotype photography

Comparing the technological progress in photography described above with the recent rise of deep learning text-to-image models presents an intriguing mind exercise. While this new generative tool has gained popularity among the general public, it faces resistance within the art community. Digital artists, in particular, express significant concern, distaste, and apprehension regarding AI-generated creative images. One can only speculate about the new styles this crisis might spawn. This situation highlights the need to develop a creative, artist-friendly, and partially generative application.

Philosophy of Modern Art

The early 20th century witnessed significant changes and advances in scientific, technological, and sociological areas. As artists no longer felt the need to precisely replicate reality in their work, they began exploring themes such as emotions, abstractions, and dreams. This period marked an era of experimentation with traditional rules and conventions, often breaking or dramatically altering them. Instead of striving for realistic color perception, artists embraced bold and vibrant colors, low contrast, monochrome, or even entirely colorless works. Similarly, rather than adhering to the rules of optics and physics, artists began to deconstruct reality, distort dimensions, and play with shapes and symbols to create visual illusions and simplify subjects.

The upcoming sections of this chapter will examine several selected artists in the context of their respective art movements. Their works will serve as crucial testing data for the application developed later in this thesis.

Abstract Art of Wassily Kandinsky [12] Wassily Kandinsky's work marked the beginning of a new era in art, where the intangible and ineffable found tangible form on the canvas. His approach to abstract art invited viewers to interpret and engage with his pieces on a deeply personal level, moving beyond literal interpretation to tap into the subconscious and spiritual realms. In his book „Concerning the Spiritual in Art“ [19] he writes:

Color is the keyboard, the eyes are the hammers, the soul is the piano with many strings. The artist is the hand that plays, touching one key or another, to cause vibrations in the soul.

Kandinsky sought to liberate artistic expression from the constraints of recognizable subject matter, instead focusing on evoking emotion and spirituality through form and color. His own experience with synesthesia, a condition where one sense is perceived in terms of another, played a key role in his artistic vision, as he often heard colors and saw sounds. This allowed him to explore the synthesis of color, line, and shape to convey profound inner experiences and universal truths, offering viewers a unique and immersive experience in his work. The main features of this style are:

- Dynamic non-representational forms
- Vibrant color palette
- Rhythmic and harmonious composition

Examples of Kandinsky's artworks can be seen in Figure 2.2.

Cubism of Paul Klee [2] Cubism emerged as a revolutionary art movement pioneered by Pablo Picasso and Georges Braque. It radically departed from traditional artistic conventions by depicting objects from multiple perspectives simultaneously, challenging the viewer's perception of space and form. The main features of this style are:

- Geometric shapes
- Fragmented forms
- Deconstruction of subjects into their component elements

Paul Klee's diverse artistic elements cannot be claimed strictly cubist, but are rather of a style that is distinctly his own. However, despite absence of cubists' analytical tendencies, there is a clear influence of this art movement in his work. His geometrical compositions have a unique blend of abstraction and whimsy, and are marked by a distinctive playfulness and vibrant color palette. Examples of Klee's artworks can be seen in Figure 2.3.

Neoplasticism of Piet Mondrian [4] Neoplasticism asserts that even the most realistic art styles fail to capture reality in its purest form. Every artist manipulates and refines their subject to render it aesthetically pleasing, engaging, and balanced. Striving to distill art to its purest form, aiming to achieve a universal harmony, neoplasticism artists are using the least to achieve the most.

The main features of this style are:

- Use of only primary colors: red, blue and yellow
- Only straight lines
- Non-representational compositions

Being the pioneer of the art movement, Mondrian's works exemplify this aesthetic. The key aspect is the balance between form, space and color. The goal is to evoke a sense of equilibrium and order through visual language of squares, rectangles, and straight lines. Just as any book can be represented by the alphabet, Mondrian's compositions in red, blue, and yellow encapsulate the essence of any oil painting. Examples of Mondrian's artworks can be seen in Figure 2.4.

Orphism of Robert Delaunay [14] Orphism, cofounded by Robert Delaunay and his wife, Sonia, was a short-lived movement that brought a fresh perspective to the artistic world. His artistic journey began with roots in Cubism, but he gradually shifted away from its monochromatic palette and angular forms towards a more vibrant and fluid approach. The main features of this style are:

- Bold overlapping shapes
- Dynamic compositions
- Interplay of light and movement

His works feature circular forms and vivid color gradients that create a sense of rhythm and energy, emphasizing the emotional impact of color and its ability to generate depth. Combining the structural aspects of Cubism with the rich hues of Fauvism, Delaunay's art is filled with musicality and dynamism. Examples of Delaunay's artworks can be seen in Figure 2.5.

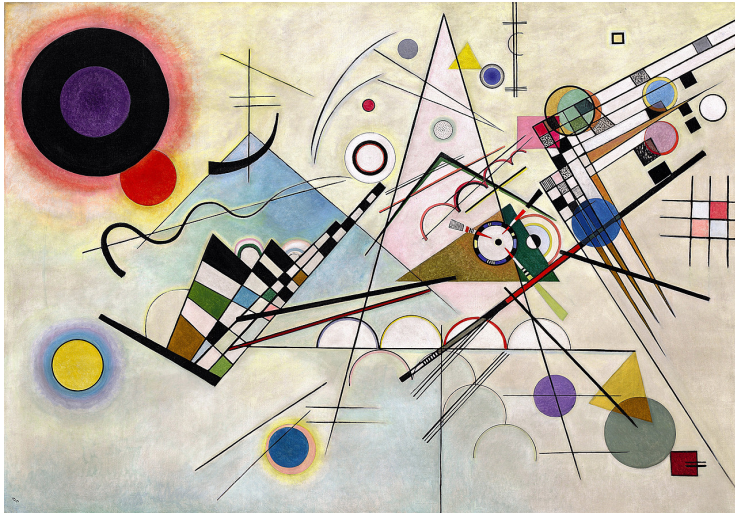
Suprematism of Kazimir Malevich [34] Suprematism, founded by Kazimir Malevich in 1913 with a series of artworks sharing the same name, was a revolutionary movement that emphasized the supremacy of feeling and perception through non-objective forms.

The main features of this style are:

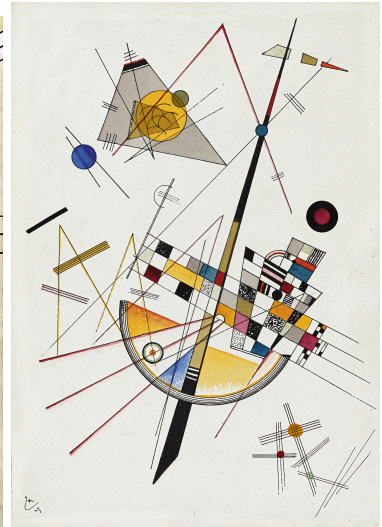
- Basic geometric shapes
- Limited range of colors
- Simplicity and reduction

Malevich's distinctive style features simple, abstract shapes such as squares, rectangles, and circles, often arranged in a seemingly floating, asymmetrical manner on the canvas. These geometric forms are rendered in a limited color palette, creating a striking visual impact that emphasises form and composition. The level of abstraction suggest movement and depth, creating a sense of weightlessness and dynamic balance. The forms and colors exist in a non-Euclidean space, allowing the paintings to be hung in any orientation, which elevates the focus from representational art to pure feeling and perception, leaving room for viewers to interpret and engage with the work on a personal level.

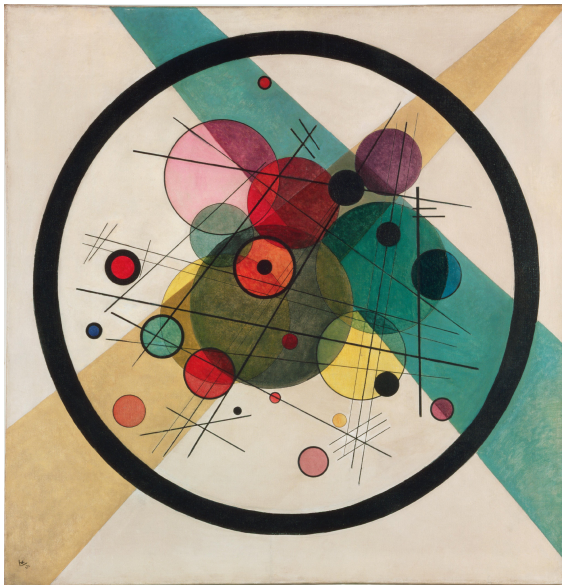
Examples of Malevich's artworks can be seen in Figure 2.6.



(a) Composition 8 [16]



(b) Delicate Tension [17]



(c) Circles in a Circle [15]

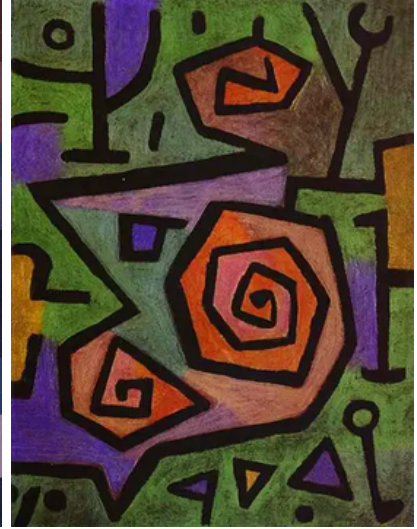


(d) On White II [18]

Figure 2.2: Works of Wassily Kandinsky



(a) Fire and Full Moon [21]



(b) Heroic Roses [23]

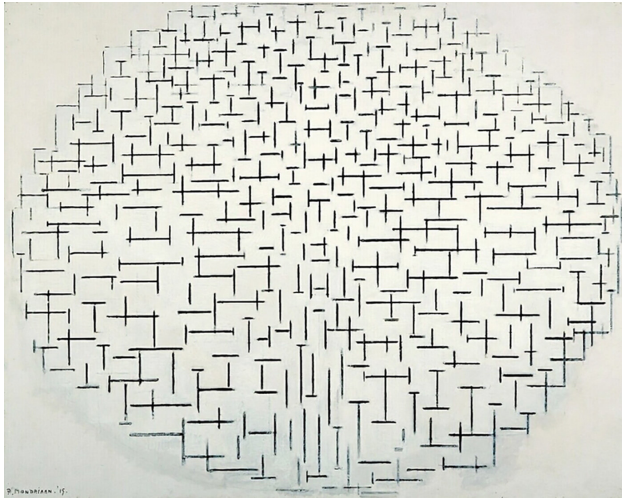


(c) Characters in Yellow [22]

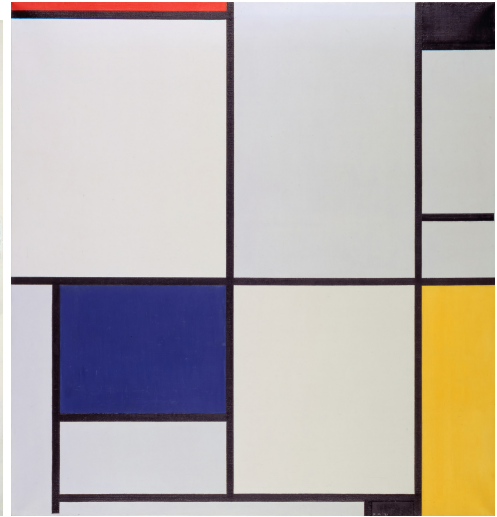


(d) Castle and Sun [20]

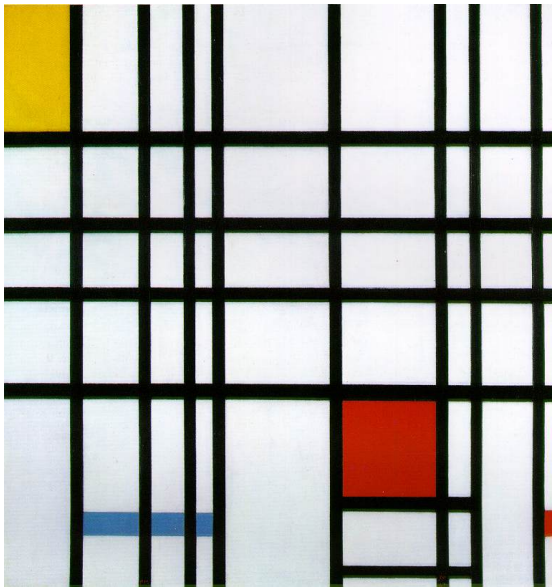
Figure 2.3: Works of Paul Klee



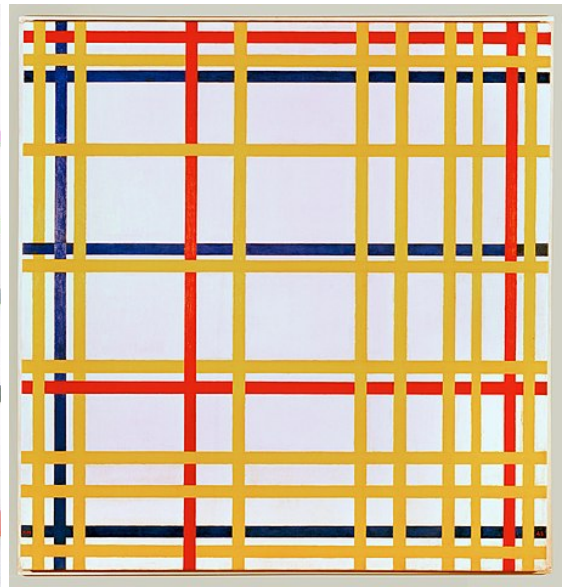
(a) Pier and Ocean [30]



(b) Tableau I [31]



(c) Composition with Red, Yellow and Blue [32]



(d) New York City I [33]

Figure 2.4: Works of Piet Mondrian



(a) Rythme [8]



(b) Circular Forms [7]



(c) Red Eiffel Tower [5]

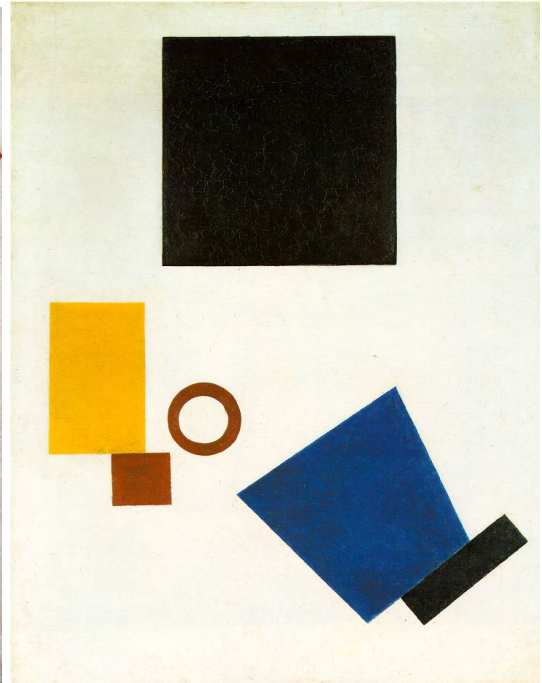


(d) Simultaneous Windows [6]

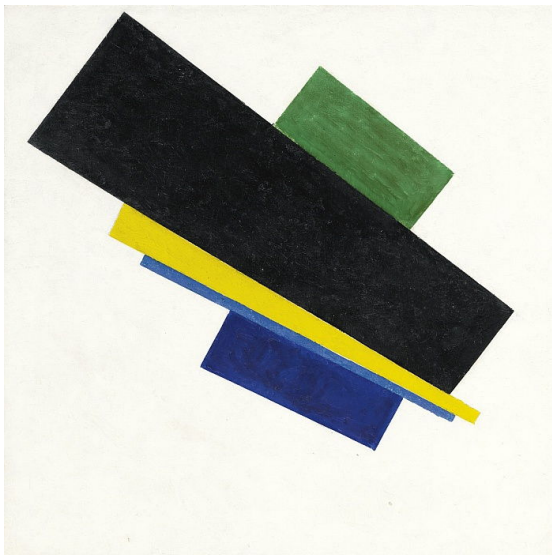
Figure 2.5: Works of Robert Delaunay



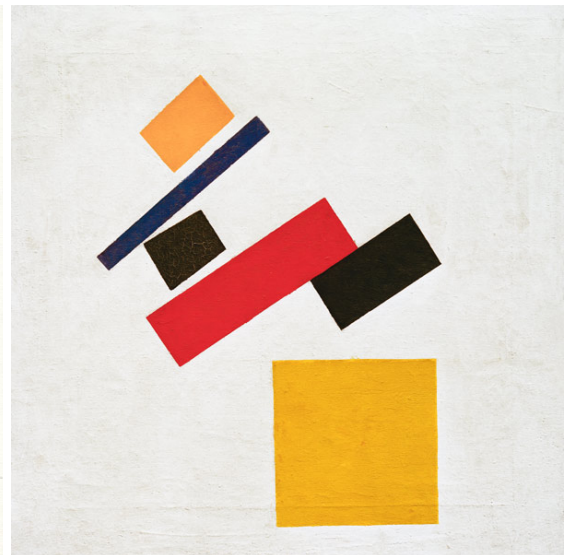
(a) Eight Red Rectangles [28]



(b) Self-Portrait in Two Dimensions [25]



(c) 18th Construction [27]



(d) Suprematism [26]

Figure 2.6: Works of Kazimir Malevich

Chapter 3

Two-Dimensional Automata

This chapter will provide formal definition and examples of translation automata and cellular automata, discuss their similarities, differences and potential use in context of visual art.

Translation Automaton

Translation Automaton (TA), also known as *Finite Transducer* or *Finite Automaton with Output* is a mathematical model of computation similar to Finite Automaton [29]. It is an abstraction of a machine that can be in exactly one state at a time and can transition from one state to another in response to input. Apart from classical Finite Automata, a Translation Automaton not only has an input, but also an output tape. Formally, it can be described as follows:

Definition 1. *Translation automaton is a quintuple $M = (Q, s, F, \Sigma, R)$, where*

- Q is a finite set of states
- $s \in Q$ is the start state
- $F \subseteq Q$ is a set of final states
- Σ is an alphabet such that $\Sigma \cap Q = \emptyset$ and $\Sigma = \Sigma_I \cup \Sigma_O$, where Σ_I is an input alphabet and Σ_O is an output alphabet
- $R \subseteq Q(\Sigma_I \cup \{\epsilon\}) \times Q\Sigma_O^*$ is a finite set of transition rules, denoted as $(pa, qz) \in R$ for $p, q \in Q, a \in \Sigma_I \cup \{\epsilon\}, z \in \Sigma_O^*$

To better illustrate the principle of TA, consider the following example. Imagine an automaton M that is designed to remove redundant unary operators of negation. That is, for an even number of negation operators (ex. $\neg\neg a$) automaton output is a and for an odd number the output is $\neg a$. Set of states is then defined as $Q = \{s, q, f\}$, where s is a start state and $f \in F$. The alphabet $\Sigma = \{\neg, a, \epsilon\}$ where ϵ is an empty symbol and $\Sigma_I = \{\neg, a\}$ while $\Sigma_O = \Sigma$. Computation begins in the state s , which corresponds to an even number of read negation symbols (treating zero as an even number), while q corresponds to an odd number of read \neg and f corresponds to read a . The rule set is then:

$$\begin{aligned}
R &= \{1, 2, 3, 4\} \\
1 &: (sa, fa) \\
2 &: (s\neg, q\epsilon) \\
3 &: (q\neg, s\epsilon) \\
4 &: (qa, f\neg a)
\end{aligned}$$

Meaning, that if symbol a is read in s state, M moves to state f and writes a to output. If symbol \neg is read, M moves to state q . Note that when \neg symbol is read, no symbol, or formally ϵ , is written to output. This holds true also in case of state q , where symbol \neg moves M back to state s . If a is read in q , the state is changed to f and sequence $\neg a$ is written.

Such an automaton is capable to accept any number of negation symbols and provide correct output. For example, to accept string $\neg\neg\neg a$, M applies the following rule sequence: 2, 3, 2, 1. For an even number of negation symbols the sequence would be 2, 3, 2, ..., 3, 4, always oscillating between states s and q until a symbol is read.

The use of translation automaton in artistic applications holds significant potential. The creation of an artwork can be conceptualized within the formal framework of TA. Initially, each piece begins in a start state, typically a blank canvas, and then transitions through various states, sometimes reverting to earlier states by undoing progress—either digitally or physically through erasing, covering, or using a new canvas. The input alphabet can be understood as the colors employed, while the output alphabet encompasses the shape, placement, and quantity of color in the artwork. The artist’s vision and style serve as the mapping rules from input to output. Finally, the set of final states represents the diverse range of variations that an artist is capable of producing.

The variety of potential outcomes that can be generated by an artist with a blank canvas is virtually limitless. For instance, Bob Ross, a well-known television artist, demonstrated remarkable productivity in his series „The Joy of Painting.“ Over the course of 11 years, he created 403 unique landscape, seascape, and winter scene paintings—one for each episode. This equates to roughly 36 paintings per year or three paintings per month, showcasing a significant amount of dedication and creativity. Although an advanced translation automaton could theoretically surpass these production rates and generate numerous artworks at once, the objective of this research is not to replace artists, but rather to support and enhance their creative process.

The previous discussion has focused on one-dimensional translation automaton. However, since the thesis aims to apply automaton theory to images, a two-dimensional approach is more suitable. The extension of one-dimensional translation automaton to two dimensions involves adding directionality to the transition rules. Rather than working with an input string that allows movement only from left to right, the automaton processes an input matrix, which permits movement in four directions. Each transition rule specifies a direction for the input head movement after the rule is applied. Formally, transition rule set can be defined as follows:

$$R \subseteq Q(\Sigma_I \cup \{\epsilon\}) \times Q\Sigma_O^* \{\leftarrow, \rightarrow, \uparrow, \downarrow\}$$

To better illustrate principle of 2D translation automaton, consider following example. A two dimensional automaton M_2 that is designed to call Czech Republic police number

(158) on a standard three by three phone number layout regardless of the starting position of the input head. The state set is defined as $Q = \{s, a, b, f\}$, where s is a start state and $f \in F$. The input alphabet consists of numbers from 1 to 9, and the output alphabet is the same but extended with empty symbol. To simplify the notation of the rules, if the same rule is applicable for several input symbols, all accepted symbols are listed in set, for example $(s\{1, 2, 3\}, f\epsilon \downarrow)$. The rule set is then defined as follows:

$$\begin{aligned}
R &= \{A, B, C, D, E, F, G, H, I, J, K, L\} \\
A &: (s\{2, 3, 5, 6, 8, 9\}, s\epsilon \leftarrow) \\
B &: (s\{4, 7\}, s\epsilon \uparrow) \\
C &: (s1, a1 \rightarrow) \\
D &: (a\{1, 4, 7\}, a\epsilon \rightarrow) \\
E &: (a\{3, 6, 9\}, a\epsilon \leftarrow) \\
F &: (a2, a\epsilon \downarrow) \\
G &: (a8, a\epsilon \uparrow) \\
H &: (a5, b5 \rightarrow) \\
I &: (b\{1, 2, 3, 4, 5, 6\}, b\epsilon \downarrow) \\
J &: (b7, b\epsilon \rightarrow) \\
K &: (b9, b\epsilon \leftarrow) \\
L &: (b8, f8)
\end{aligned}$$

The input matrix looks as follows:

$$\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}$$

Consider the starting position to be at the right bottom corner. The input symbol is 9 and the state is s . According to the rule set, M_2 is going to apply rules in sequence $A, A, B, B, C, F, H, I, K, L$ and will end up with the string „158“ on the output successfully. This simple example is very specific to showcases the principle of the rule application and have a strictly specified input by design. Other use cases are more general, for example pattern recognition or edge detection are possible.

By choosing an appropriate design, an art processing 2D translation automaton can be constructed. Another possibility is the use of a similar concept, a cellular automaton, which offers a more practical application of automaton theory in two dimensions. This method provides a smooth transition to the study of automata in image processing and aligns well with the goals of the thesis, as further explained in the next section.

Cellular Automaton

Cellular automaton (CA) is a discrete computational model inspired by the functioning and communication of living cells. It consists of *cells* organized in a lattice, where each cell can be in one of several possible *states*. The state of all cells at a particular moment is referred to as the CA *configuration*. Computation takes place in discrete steps, known as

iterations, where a cell's new state depends on the states of nearby cells. The specific set of surrounding cells that influences the computation is called the *neighborhood*, defined by the position of cells relative to the active cell. The relation between the configuration of the neighborhood and the resulting state is known as the *CA rule*.

By applying the rules, new states are computed simultaneously for each cell, and the updated configuration is assigned to the cellular automaton with each iteration. Significant variations arise from differences in the size of the state set, specific rules, and various types of lattices. Lattices can be finite or infinite, uniform or non-uniform, and exist in one, two, or higher dimensions. The lattice properties influence the shape of the cells, which in turn determines the potential neighborhood configurations. Commonly used lattices include uniform ones, sometimes with triangular[40] or hexagonal[41] shapes, but most frequently using square tessellations. A square cell can have up to eight immediate neighbors, with the most common neighborhood types being Moore and Von Neumann, as seen in Figure 3.1. Both of these are static, but there are also dynamic neighborhoods such as the Margolus neighborhood 3.2, which alternates at each step.

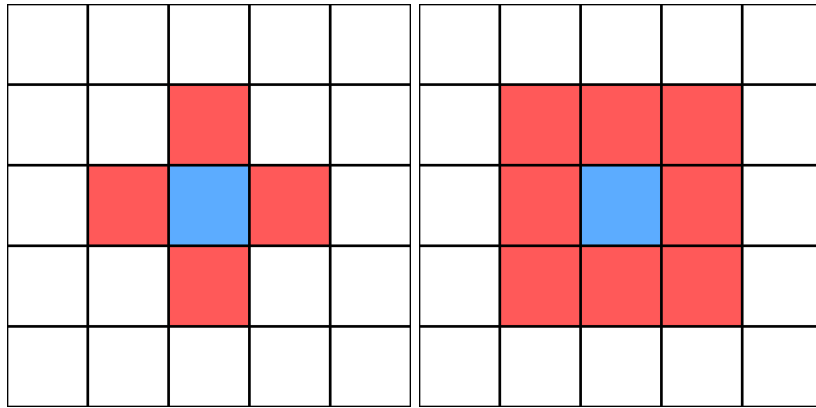


Figure 3.1: Neighborhood models of Von Neumann (left) and Moore (right)

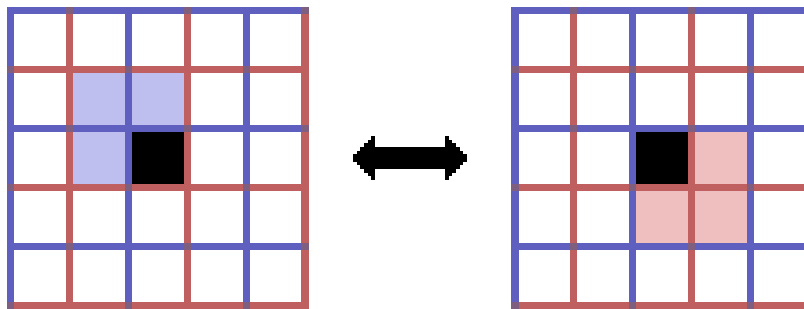


Figure 3.2: Margolus neighborhood, also known as block cellular automaton

Closely connected to the neighborhood aspect of the CA is the boundary condition. In a finite lattice, there is always a subset of cells at the border that lack at least one defined neighbor. To calculate a new state for these cells, certain constraints must be applied. Options include assigning a constant state to the undefined cell, duplicating the value of the active cell, or leaving the border cell unchanged and only using it as a neighbor. A particularly beneficial option is the periodic boundary, where undefined neighbor beyond the border is assigned with the value of the border cell from the opposite side. This can

be topologically perceived as creating a loop for a 1D lattice or a torus for a 2D lattice. A visual representation of this boundary condition is shown in the figure 3.3.

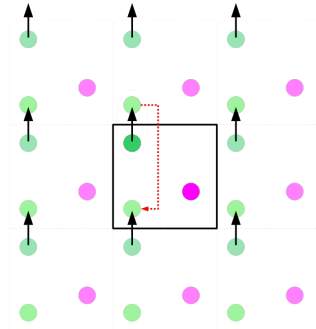


Figure 3.3: Periodic boundary graphic representation in 2D

The size of the neighborhood and the range of possible cell states are the primary factors influencing the complexity of cellular automata. The simplest form of CA is known as Wolfram’s CA, which features a 1D configuration with square cells, two possible states, and a neighborhood consisting of three cells: an active cell and its right and left neighbors. This setup results in eight possible neighborhood configurations (2^3) and consequently allows for 256 potential rules (2^8). As the complexity of the automaton increases, the number of possible rules rises exponentially. For instance, by modifying Wolfram’s CA to use three states, the possible rules would increase to 19,683 (3^9). Additionally, increasing the neighborhood size to include five cells results in a total of 2^{32} possible rules.

Even with such a simple automaton the creative potential is apparent. Given the relatively small number of possible rules, Wolfram’s CA has been extensively studied and its rules have been classified by Wolfram himself into the Four Classes of Behavior[39]. By stacking configurations from successive iterations one below the other, distinct patterns emerge. The rule is classified based on the pattern it produces: Class One for uniform final patterns, Class Two for alternating patterns, Class Three for random patterns, and Class Four for a combination of the characteristics from Classes Two and Three. Examples of each class can be found in Figure 3.4. From a creative perspective, Class Four rules are arguably the most valuable due to their blend of order and complexity in the patterns they generate.

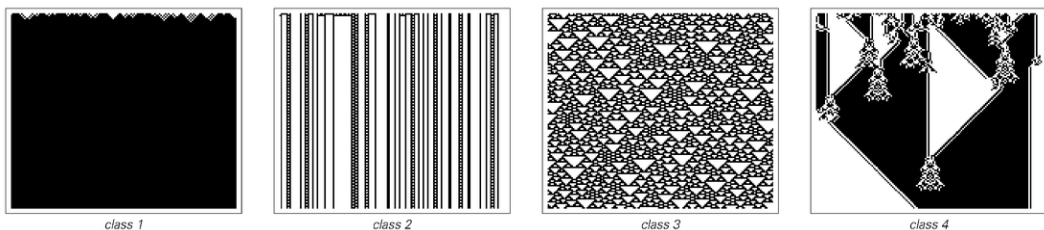


Figure 3.4: Example of the Wolfram’s Four Classes of Behavior

Exploration of an iteration stacking technique naturally transitions from pure one dimensional to actual two dimensional cellular automata. The most well-known and widely studied 2D automaton is Conway’s Game of Life (GoL) [10], which features an infinite square lattice, two states („live“ and „dead“), and the Moore neighborhood. The rules in

GoL depend solely on the total number of living cells in the neighborhood, which is why they are often referred to as totalistic rules. The rules govern state transitions: a change from a dead to a living cell is called birth, while a change from a living to a dead cell is called death. When a cell remains alive in the next iteration, it is referred to as survival. The rules are as follows:

- Birth: Exactly three neighbors must be alive.
- Survive: Two or three neighbors must be alive.
- Death: Less than two or more than three neighbors are alive.

By adjusting the numbers of required living cells in the GoL rules, a whole spectrum of related automata, often called Life-like automata, can be explored. Experimenting with Life-like automata is a field of study with its own notation and terminology. The Life-like rule is expressed using a notation called *rulestring*, with the most common format being birth/survival in the form of B{number list}/S{number list}. According to this notation, the original GoL rules can be described by the rulestring B3/S23. The application of totalistic rules can generate a vast array of captivating patterns depending on the initial configuration. These patterns, which may be oscillating, moving, or static, are known as *ashes*.

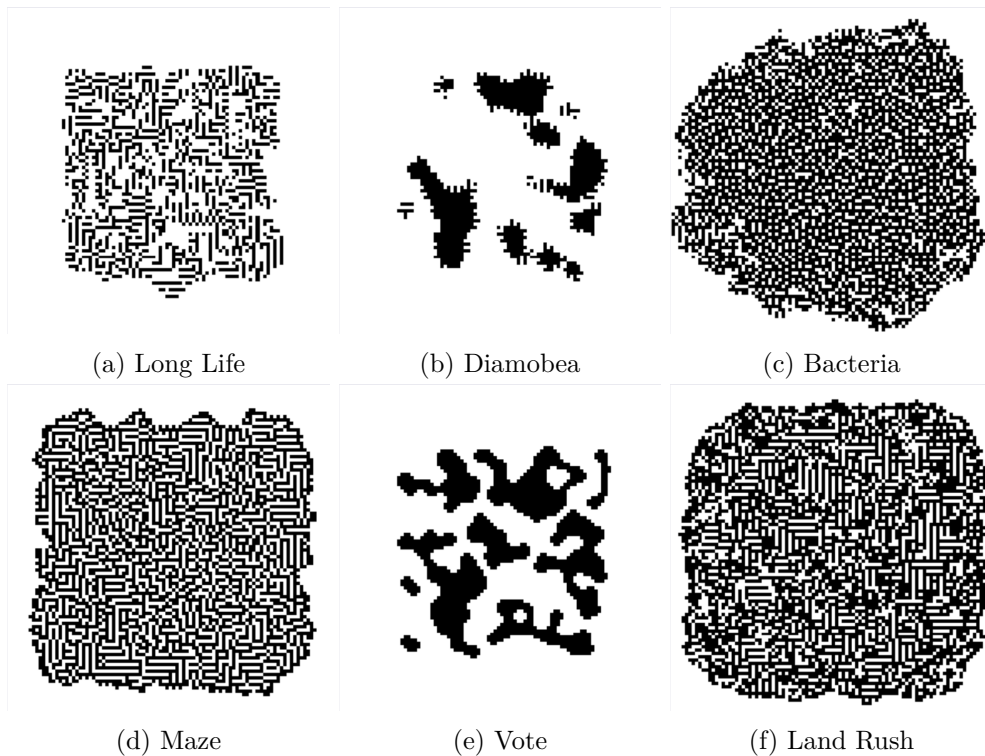


Figure 3.5: Static examples of selected totalistic rules.

The most visually appealing totalistic rules, from the perspective of generating stable images, are those categorized as stable or exploding, some are shown in Figure 3.5. These rules often produce satisfying static results even from random initial configurations. For chaotic rules and oscillating ashes, compelling visuals can be achieved by adding an *envelope*,

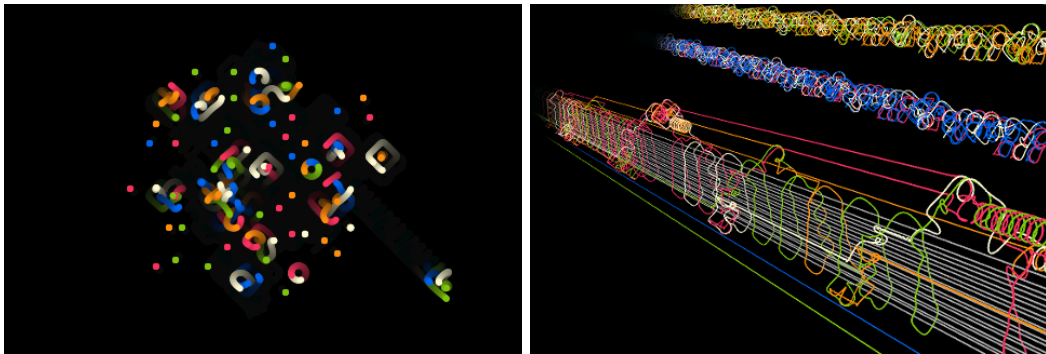
which highlights cells that were alive in at least one iteration. Additionally, many of the oscillating ashes feature intriguing shapes and may exhibit symmetries, evoking natural patterns or even slavic ornaments.

Using the Moore neighborhood and two cell states, there are 512 (2^9) possible neighborhood configurations for Life-like cellular automata, resulting in 262,144 (512^2) possible rules. Although some rules are essentially identical, the creative potential of totalistic rules is clear, and only a small portion of them have been thoroughly studied. However, totalistic 2D cellular automata represent just a subset of possible CA variations. Expanding the scope by incorporating more states, altering the neighborhood, or changing the shape of cells adds greater diversity and increases the potential to discover more valuable rules exponentially.

An intriguing example of an alternative cellular automaton is the Single Rotation (SR) rule, introduced by Dmitry Shintyakov [35]. This 2D automaton uses the Margolus neighborhood and operates with two states. Its block transition rule is straightforward:

Rotate clockwise by 90° every block with exactly one live cell.

Despite its simplicity, this rule can produce images of remarkable complexity when visualised correctly, as seen in Figure 3.6. By interpolating the movement trajectories of live cells and differentiating cells with distinct colors, abstract geometric patterns emerge that resemble the artistic style of Pollock. This allows for the creation of intricate and visually captivating designs.



(a) Two dimensional SR, taken from [36] (b) Three dimensional SR, taken from [37]

Figure 3.6: Visualisation of Single Rotation rule with frame interpolation

Application Opportunities

When designing the application, various aspects must be considered. Cellular automata have already seen widespread use in image processing tasks such as noise reduction, edge detection, and texture recognition, with numerous effective CA rules described in the literature. Reusing and extending these rules can simplify the implementation process. Translation Automaton can be equated with Cellular Automaton by treating a CA configuration as a state of the TA. In this context, the input alphabet consists of all possible neighborhood configurations, and the output alphabet aligns with the set of cell states. This mapping allows for a direct correlation between CA and TA rules.

Combining different CA rules can significantly expand the range of possibilities and provide users with insightful and inspiring ideas. Allowing the user to choose the number

of iterations and other parameters ensures interactivity and variation in the results. In the future, this manual element can be automated through metadata randomization and the application of evolutionary algorithms.

In the context of cellular automata research, evolutionary algorithms are commonly employed to optimize a given task by encoding and mutating CA rules. This approach can be taken a step further by not just searching for one optimal rule but instead exploring combinations, orders, and parameters that yield the most visually appealing outcomes. These aspects will be further detailed in the following chapter.

Chapter 4

Architecture and Features

In Chapter 2, significant features of geometric and abstract art were explored. In Chapter 3, the principle of translation cellular automata and their potential utilization were examined. This chapter aims to apply this knowledge in the future development of an application that supports and promotes creative flow and experimentation for users.

Structure of the App

The expected workflow for the potential user involves the following steps: uploading an image, experimenting with various rules and additional features, and then downloading the final result. Consequently, the interface must offer clear options for uploading and downloading images, as well as a section dedicated to rule addition and parameterization. It is also beneficial to provide the user with the ability to undo their most recent action. To enhance the user experience, the application should include several built-in default images.

The React framework was utilized for the implementation due to its ease of access, deployment, and development capabilities. The application's main components include the Canvas, which facilitates image upload, download, and display; Quantization, which is discussed in detail later; and the Transformer, responsible for the actual image editing process.

To effectively combine different rules, a RuleList component is introduced as part of the Transformer. This component enables adding, deleting, and disabling rules and allows users to clear the list using a single „delete all rules“ button. A dedicated Rule component handles rule customization. The layout of each rule is determined by its type, with a customized representation based on the specific parameters required for the rule to operate. Rule metadata includes a `complete` flag indicating whether all parameters have been set for the rule. This flag is hidden from the user and serves to exclude unfinished rules during the transformation process.

The RuleList state, along with a default palette and a list of available rule types, are stored in local storage for persistent state management. Upon completing each transformation, the outcome is recorded in history with the help of a custom `useHistory` hook, enabling users to perform undo and redo actions.

Although the rules' representation and the transformation process are integrated within the Transformer component, they are kept logically and structurally distinct. Communication between these parts is handled via rule metadata. This separation allows the transformation logic to be completely isolated and potentially reimplemented for specific hardware configurations.

Automatic Rule Generation

As briefly discussed in 3, stochastic generation of rules can be added to the application not only to promote creativity and exploratory nature of art, but also as a base for future implementation of an evolutionary algorithm.

Given the predefined list of available rule types and their corresponding parameters, it is feasible to generate rules at random. The most straightforward approach involves setting all possible parameters, regardless of the rule type. With this approach, simply adding new rules to the RuleList is enough for proper display in the user interface, as rule representation is based solely on the rule metadata. Furthermore, when employed as a genome in an evolutionary algorithm, such rules are easy to modify through crossover and mutation. While this method of rule generation is generally effective, it may become less manageable if the list of possible parameters becomes too extensive, in which case a more advanced approach may be required.

Aesthetic Quality Metrics

Providing artists with aesthetic metrics for the final image, whether generated or manually adjusted, can aid in evaluating the quality of the outcome. Additionally, these metrics could later serve as fitness functions in an evolutionary algorithm. The most common metric for assessing artistic quality is the balance between redundancy and entropy, also known in literature as the contrast between unity and variety, or order and complexity.

The redundancy metric is calculated based on the informational value of the image, which is determined by the proportional representation of individual colors [13]. Given an image with m colors and N pixels, the sum of pixels for each color can be calculated, resulting in a set of sums:

$$\{n_1, \dots, n_m\} \text{ where } n < N \text{ and } \sum_{i=1}^m n_i = N \quad (4.1)$$

The individual probability for each color is calculated as follows:

$$P_i = n_i/N \quad (4.2)$$

Given the probability P_i of each color i in the image, the entropy of the color can be calculated using the following formula:

$$H_i = -\log_2(P_i) \quad (4.3)$$

To calculate the overall entropy of an image, a weighted sum of the entropies of its individual colors is computed:

$$H = \sum_{i=1}^m P_i \times H_i \quad (4.4)$$

To calculate redundancy, the maximum entropy must first be determined. Maximum entropy occurs when each color in the image has an equal probability:

$$H_{max} = -\sum_{i=1}^m pp_i/N \times \log_2(pp_i/N) \quad (4.5a)$$

$$\text{where } pp_i = \text{round}(N/m) \text{ so that } \sum_{i=1}^m pp_i = N \quad (4.5b)$$

To calculate redundancy, divide the actual entropy by the maximum entropy, and then subtract this value from one.

$$R = 1 - (H/H_{max}) \quad (4.6)$$

This only works if the value of H_{max} is not equal to zero, as it can happen if there is only one color. If the max entropy is zero, the value of redundancy automatically equals to one.

A redundancy value closer to zero indicates that the image is maximizing the information it can contain. This could manifest as white noise or a perfectly balanced two-tone image. Conversely, a redundancy value closer to one implies minimal information content. Examples include a single pixel-thick black circle on a white background or a simple vector image of a plant.

The pixel positioning is irrelevant in this context; only the colors and their distribution are assessed. Evaluation of the used dataset suggests that aesthetically pleasing images generally have redundancy values in the range of 0.3 to 0.7.

This algorithm is not dependent on the image’s palette; however, it is important to note that color quantization, which involves reducing the number of colors in an image, can significantly impact the outcome of this metric.

While redundancy is an intriguing theoretical concept and can be somewhat useful in practice, it alone is insufficient to fully assess the artistic quality of an image. Important traditional metrics such as color harmony, composition, symmetry, negative space, and line of action play crucial roles in evaluating the visual appeal of an image and may be considered for inclusion in future versions.

In subsequent developments, combining rule generation with artistic metrics could facilitate the implementation of a genetic algorithm that automatically produces various edited versions of an image. Ideally, this generative process would optimize results based on selected metrics while simultaneously preserving the value of other metrics.

Utilization of Cellular Automaton

Before performing any transformations, the cellular automaton must be conceptualized. This is achieved in the application through the following mapping:

- The *finite grid* is represented by the image uploaded by the user.
- The *cell* corresponds to a pixel within the image.
- The cell’s *state* is defined as the color of the respective pixel.

Several considerations need to be addressed before adopting this model, one of which concerns the boundary condition of the finite grid. As discussed in 3, there are several options available, but the most appropriate for this application is the periodic boundary condition due to its minimal restrictions and suitability for this context.

The next consideration is the cell’s state. Since computations are inherently discrete, it is preferable to have a discrete state space for the cell. However, because the cell’s state is determined by its color and the color space is continuous, the state space is continuous as well. The most common digital color representation is RGB coding, which uses 8 bits per channel. Although this discretization provides 256^3 possible colors (or cell states), equating to over 16 million different shades, the human eye cannot differentiate all of them. Thus, it

is feasible to use approximation to reduce the number of states and simplify computations without significant perceptual loss.

To achieve this, the application uses the K-means algorithm as part of the Quantization component. This algorithm is chosen for its widespread use in color quantization. In future versions of the app, alternative algorithms, ideally non-parametric ones, may be explored. To integrate K-means into the application, the „ml-kmeans“ npm package [1] was used. Since the data from the Canvas component is formatted differently from the required input array of [r, g, b], the data must first be rearranged. Once the data is formatted, it is processed by the library, which returns clusters and centroids. The clusters represent the colors for the resulting imagePalette data structure, which is then displayed to the user as the outcome of the quantization process. This approach provides the user with a ready-to-use palette, enhancing productivity and supporting a smooth workflow.

Even though quantization reduces the initial number of states, the transformation process may introduce new colors and shades. It can be helpful to compare colors based on their similarity. To achieve this, the application uses the Manhattan distance metric. Also known as the city block metric, Manhattan distance is a specific case of the Minkowski distance metric [24], defined as follows:

$$M(p, a, b) = \left(\sum_{i=1}^n |a_i - b_i|^p \right)^{1/p} \quad (4.7)$$

In the case of Manhattan distance, the Minkowski distance parameter p is set to one. This metric measures the distance between two points in a grid-like path, moving at right angles. It is chosen for its simplicity, as it avoids the need to calculate powers or roots, which would be required if the p parameter were greater than one. The formal definition of Manhattan distance is as follows:

$$M(1, a, b) = \sum_{i=1}^n |a_i - b_i| \quad (4.8)$$

Where a, b are points in the color space and $n = 3$ as we have three parts of the color coding.

Adding Cellular Automaton Rules

The final task is to determine the rule set itself. This involves choosing which rules to implement, their order of execution, and how they interact with each other and with user inputs. By carefully selecting and combining rules, the application can achieve a wide range of creative transformations while maintaining user control and customization options. The following chapter will delve into addressing these challenges, exploring potential rule combinations and their effects on image processing and artistic outcomes.

Chapter 5

Experimenting with the Rule Set

In Chapter 4, a cellular automaton for the application was established, addressing various challenges and incorporating several useful additional features. This chapter focuses on selecting, adapting and creating specific cellular automaton rules, combining and experimenting with them to produce visually appealing results.

Color Change Changing colors is a fundamental function in image editing software. The color change rule can be defined as follows:

If a cell's state is A, change it to B; otherwise, keep the state unchanged.

Given the simplicity and frequent use of this rule, it may be included several times in a rule list. Processing each color rule separately can lead to inefficiencies, as each pixel must be tested repeatedly for each rule. Moreover, sequentially applying rules to swap colors (e.g., from A to B and B to A) can cause unexpected behavior due to the order of execution.

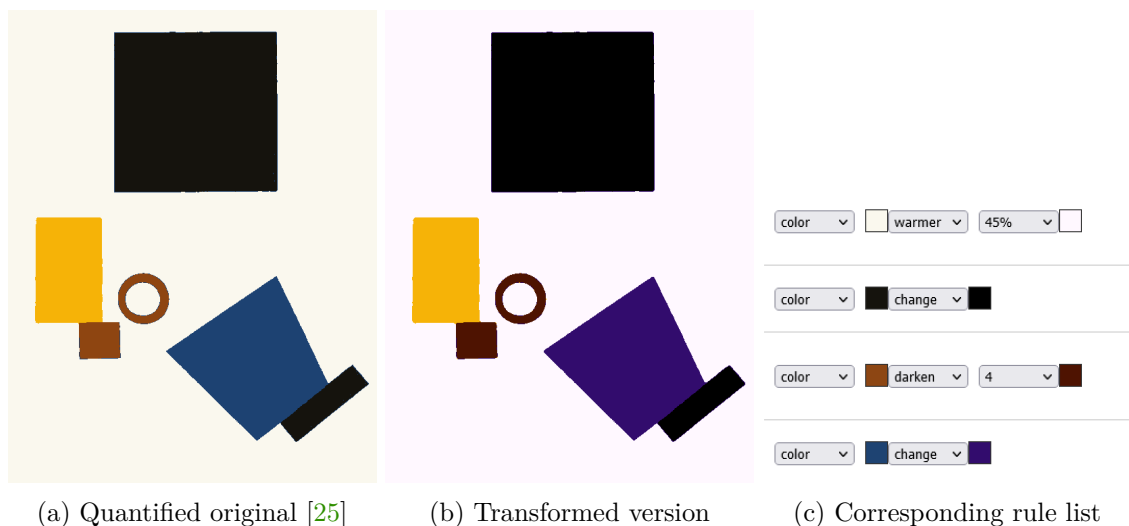


Figure 5.1: Color rule in practice

To address these issues, all color rules are applied in a single iteration, allowing each pixel to be tested only once, regardless of the number of color rules. During the transformation step, all rules are processed before any changes are applied. Each rule's original and

transformed color pairs are stored in a JavaScript object, with the original color’s hex value as the key and the transformed color as the value. After processing, the hex value of each pixel is checked against the object; either the corresponding color is retrieved or the pixel is left unchanged if no matching rule is found. This approach ensures that when multiple rules use the same original color, only the most recent rule is applied for transformation. An example of an application of several color rules is shown in Figure 5.1.

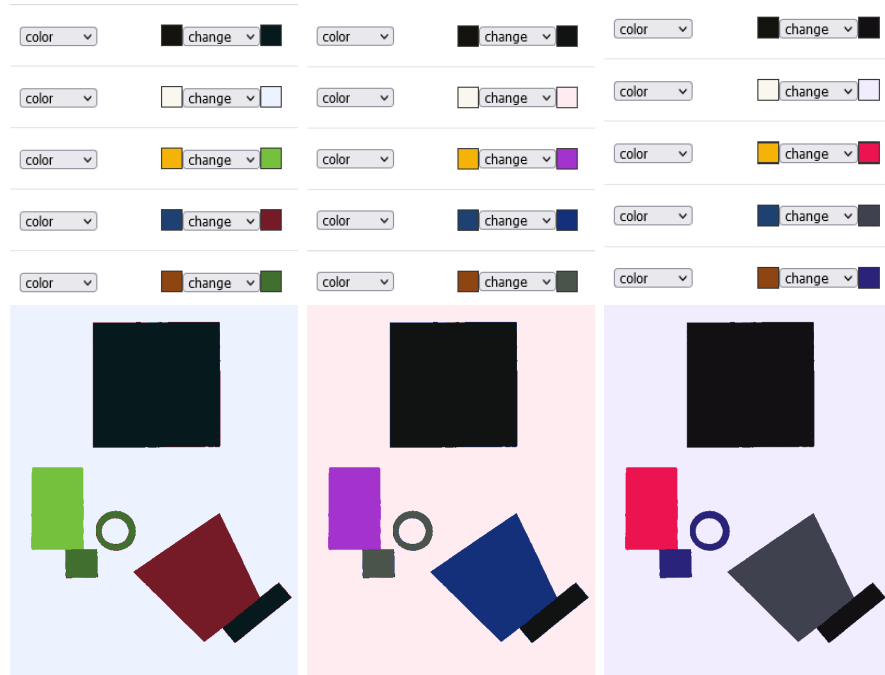
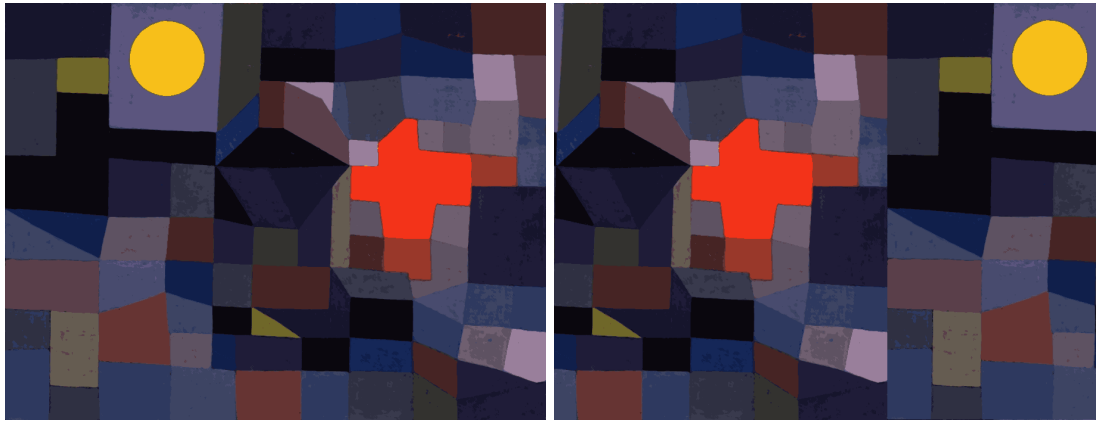


Figure 5.2: Automated generation of recoloring rules from image palette

The user can specify an original color by selecting it from the default palette or from the image’s existing palette. Alternatively, the ChromePicker tool within the user interface allows users to select a custom color. Manually setting and experimenting with colors can be complex and time-consuming. To streamline this process, there is a button labeled „Anti Gray Scale“ located above the rules list, which, when activated, generates a recoloring map.

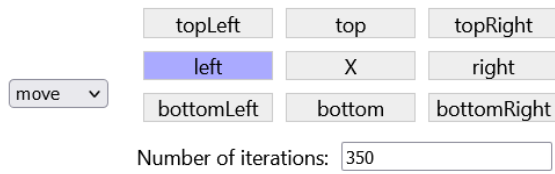
The algorithm works by converting each color in the current image palette from RGB to HSL color space, then randomly altering the hue and saturation values while maintaining a consistent lightness level. After these adjustments, the color is converted back to RGB. This recoloring map is used to create new rules that user can modify before applying them to the image, as illustrated in Figure 5.2.

Linear Translation The definition of the linear translation rule is sourced from [3]. In this rule, boundary conditions are of significant importance. As discussed in Chapter 4, periodic boundary condition is implemented in the cellular automaton. This effectively extends the finite grid into an infinitely repeating space, enabling unrestricted movement in any direction. As illustrated in Figure 5.3, for certain images, this transformation can be nearly seamless. The definition of this rule is sourced from [3] and can be written as follows:



(a) Quantified original [21]

(b) Left shift for 350 pixels

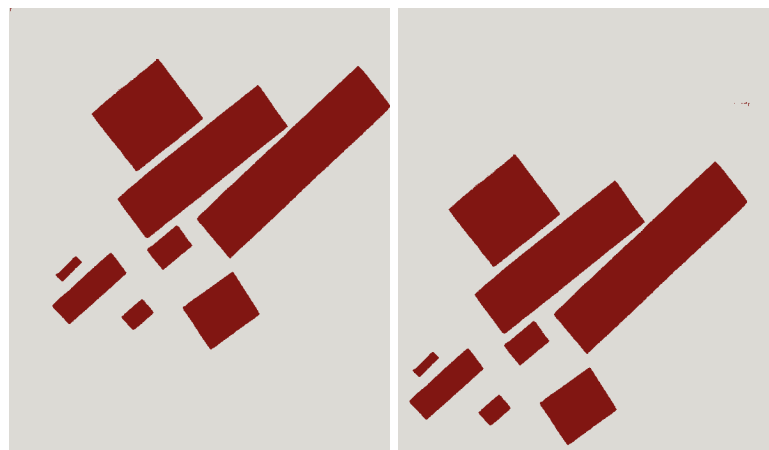


(c) Corresponding UI rule representation

Figure 5.3: Move rule in practice

Copy the state of the neighbor from the direction opposite to the one selected.

While the rule may seem counterintuitive at first, it is quite simple in practice. Copying the top neighbor's state shifts the image downward, while copying the right neighbor's state shifts the image to the left, and so on. Consequently, the rule's metadata contains two sets of directions: one for visual representation and another for actual rule application. Adjusting the direction is straightforward and user-friendly, enabling users to combine multiple move rules to shift the image in any desired complex direction, as is illustrated in Figure 5.4.



(a) Quantified original [28]

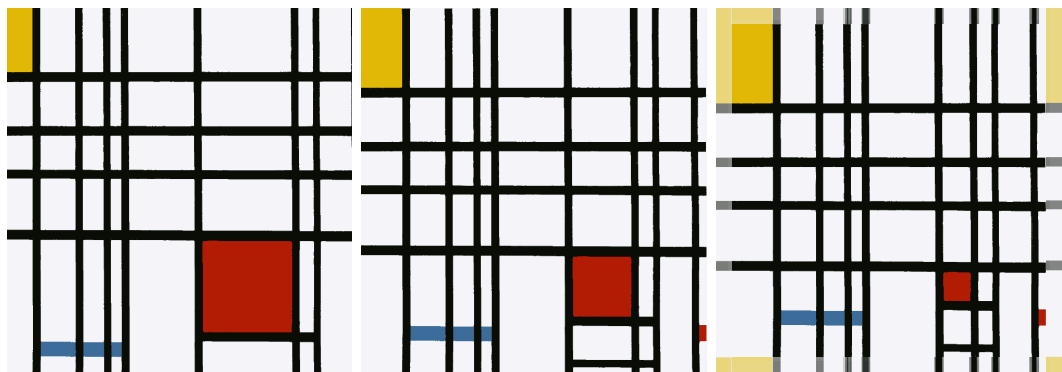
(b) Transformed image

Figure 5.4: Several move rules with different directions applied in sequence

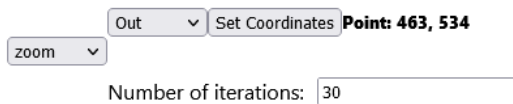
Zoom In and Zoom Out This rule exemplifies a hybrid cellular automaton where different rules apply to different cells. The image is virtually divided into four segments using one vertical and one horizontal line, with the center point serving as the intersection of these lines. The „center point“ can be either automatically computed or manually chosen by the user. Depending on the sector to which a given cell belongs and the zoom mode (In or Out), different rules are applied. The definition of this rule can be written as follows:

Determine the sector of the cell. If the mode is zoom in, move the cell in the direction of the sector, thereby moving away from the center point. If the mode is zoom out, move the cell in the direction opposite to the sector, thereby moving towards the center point.

As described in 5, moving a pixel in any direction requires copying the value from the opposite direction. This holds true for this rule as well. The rule can be further expanded by either using diagonal lines in place of vertical and horizontal lines or by combining both sets of dividing lines, which results in an increased number of sectors. It should be noted that this rule underwent significant modifications during experiments and is not a literal implementation of the original rule, which can be found in [3].



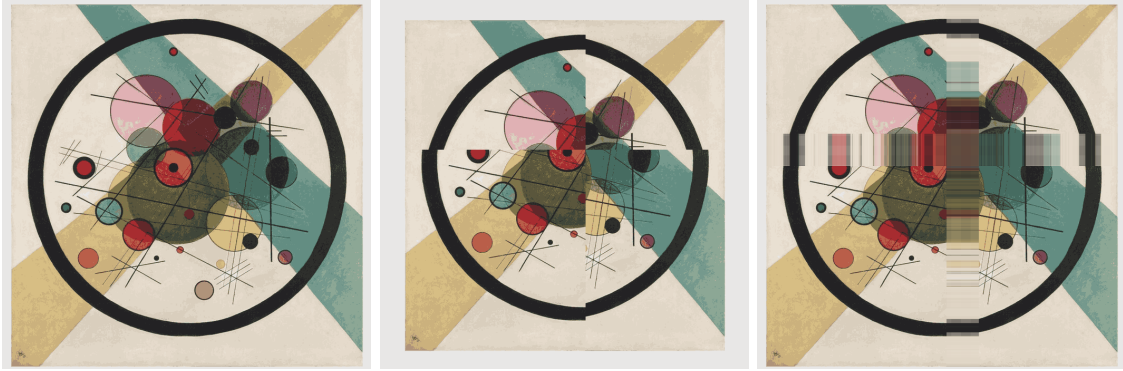
(a) Zoom In transformation (b) Quantified original [32] (c) Zoom Out transformation



(d) User interface representation of the rule

Figure 5.5: Zoom rule in practice

The effect produced is not a perfect replication of zooming in the strict sense, though it can resemble zooming for square-shaped objects, as seen in Figure 5.5. However, this rule can be applied creatively to any image. A particularly noteworthy outcome of this rule is the artifact that arises from repeating negative zoom actions with an equal number of iterations. This effect is demonstrated in Figure 5.6.



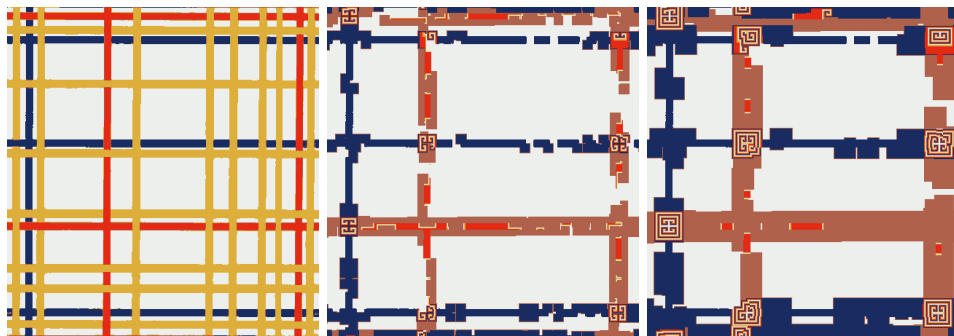
(a) Quantified original [15] (b) Zoom Out, 30 iterations (c) Zoom In, 30 iterations

Figure 5.6: Zooming artefact on non-square objects in an asymmetrical composition

Cyclic CA Developed by David Griffeath [9], cyclic cellular automaton enhances the standard cellular automaton model by introducing a complete modular ordering of cell states. Within the user interface of the application, this ordering can be customized by the user using the image palette as the foundational set. The definition of cyclic rule is as follows:

Determine the cell's current state. Obtain the successor state according to the complete modular ordering. If at least one neighboring cell is in the successor state, assign the successor state to the active cell.

When this rule is applied to an image, certain colors may be entirely eliminated as they are consumed by their successor colors. Additionally, with an appropriate ordering, sufficient entropy, and enough iterations, the cyclic rule can generate oscillating patterns similar to the Belousov-Zhabotinsky reaction. Both of these effects are illustrated in Figure 5.7.



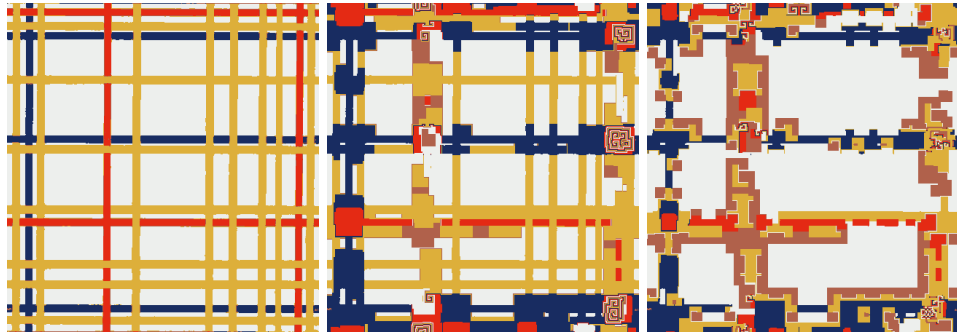
(a) Quantified original [33] (b) Transformed with 8 iterations (c) Transformed with 16 iterations



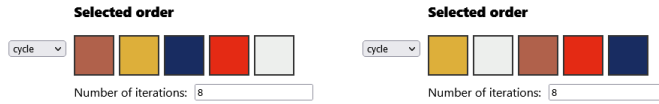
(d) UI for order selection (e) UI representation of the rule

Figure 5.7: Cyclic rule in practice

The color order is the most crucial parameter for this rule, and it can be easily modified without having to create a new rule. When two or more different orders are applied consecutively, intriguing effects can be produced, allowing for experimentation and potentially generating complex and visually striking patterns. These results are demonstrated in Figure 5.8.



(a) Quantified original [33] (b) First transformation with sequence A (c) Second transformation with sequence B



(d) Sequence A

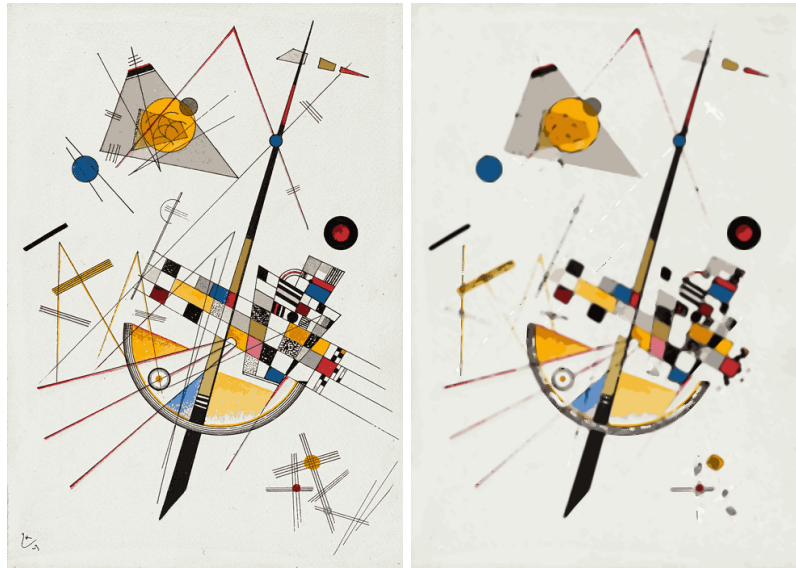
(e) Sequence B

Figure 5.8: Example of sequential change

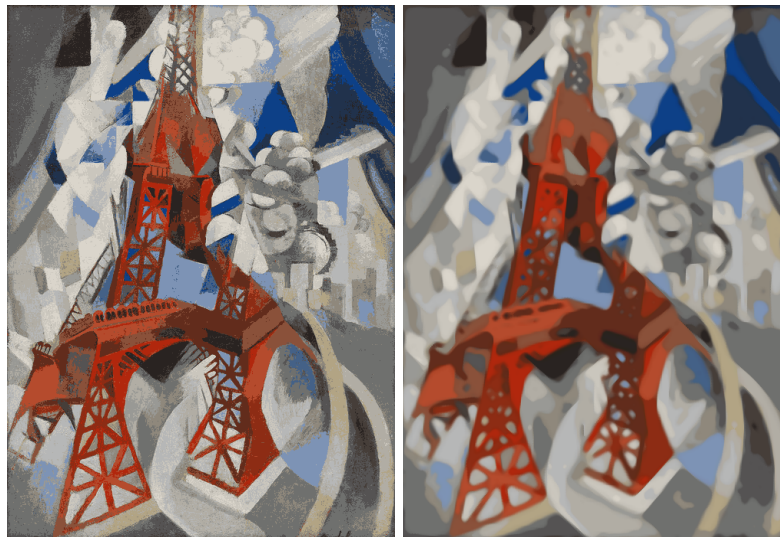
Box Blur To understand this rule, it is important to familiarize oneself with the concept of Manhattan distance, as described in 4. The definition of this rule is taken from [11] and can be written as follows:

For each pair of opposite neighbors, calculate their Manhattan distance. Select the pair with the smallest Manhattan distance. Compute the average of the chosen pair, and use this average as the next state for the active cell.

This rule is effective in simplifying image by minimizing intricate details. By smoothing out the variations in pixel values through averaging, the rule can blur or blend image features, resulting in a visual effect close to traditional blending in painting, as shown in Figure 5.9.



(a) Quantified original [17] transformed with 7 iterations



(b) Quantified original [5] transformed with 17 iterations

Figure 5.9: Box Blur rule in practice

Median Mold The Median Mold rule, similarly to the Box Blur was taken from [11] and can also be employed to smooth out image details (see fig. 5.10), but it functions differently and yields a distinct outcome. The definition of this rule is as follows:

Select three random neighbors and compute their Manhattan distance relative to the active cell. Rank the neighbors according to their distance and choose the median neighbor. Assign the state of the chosen neighbor to the active cell.

The number of randomly selected neighbors initially appeared to be arbitrary, presenting an opportunity for parameterization. To assess the effect of varying the number

of neighbors, a series of experiments were conducted. Increasing the number reduces the prominence of the molding effect; when using five or more neighbors, the effect becomes almost imperceptible and requires an excessive number of iterations to achieve noticeable changes. When the number of neighbors reaches its maximum, the effect of the rule closely resembles that of a classical median filter. Conversely, using only two neighbors resulted in excessively chaotic and destructive outcomes. Based on these findings, the decision was made to maintain the use of three neighbors for optimal performance.

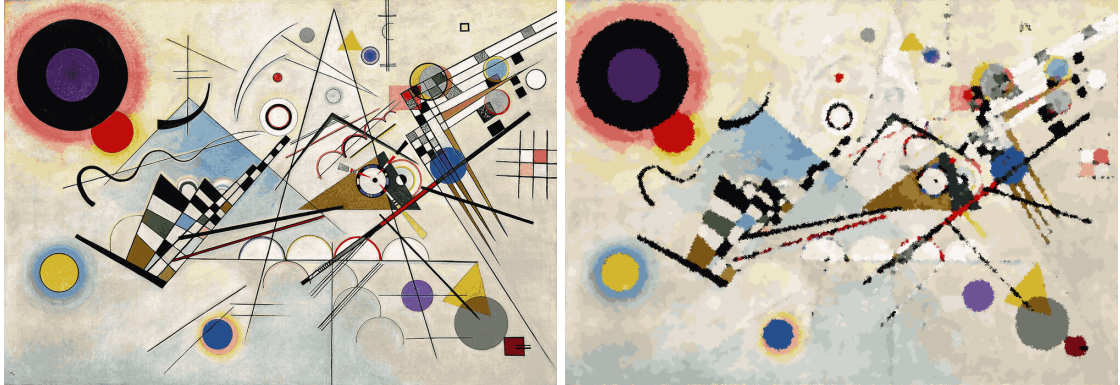


Figure 5.10: Median Mold rule in practice. Quantified original [16] transformed with 16 iterations

This rule can be applied to simulate traditional artistic materials on a canvas, particularly mimicking the texture of dry brush strokes. This effect is most apparent in paintings that are fully covered with vibrant colors and lack solid background areas, as demonstrated in Figure 5.11. Another notable characteristic of the median mold is its ability to soften sharp edges and dissolve thin lines, creating a more fluid and blended visual experience. This effect is illustrated in Figure 5.12.



(a) Quantified original [23] after 16 iterations (b) Quantified original [6] after 16 iterations

Figure 5.11: Visual effects of a brushstroke

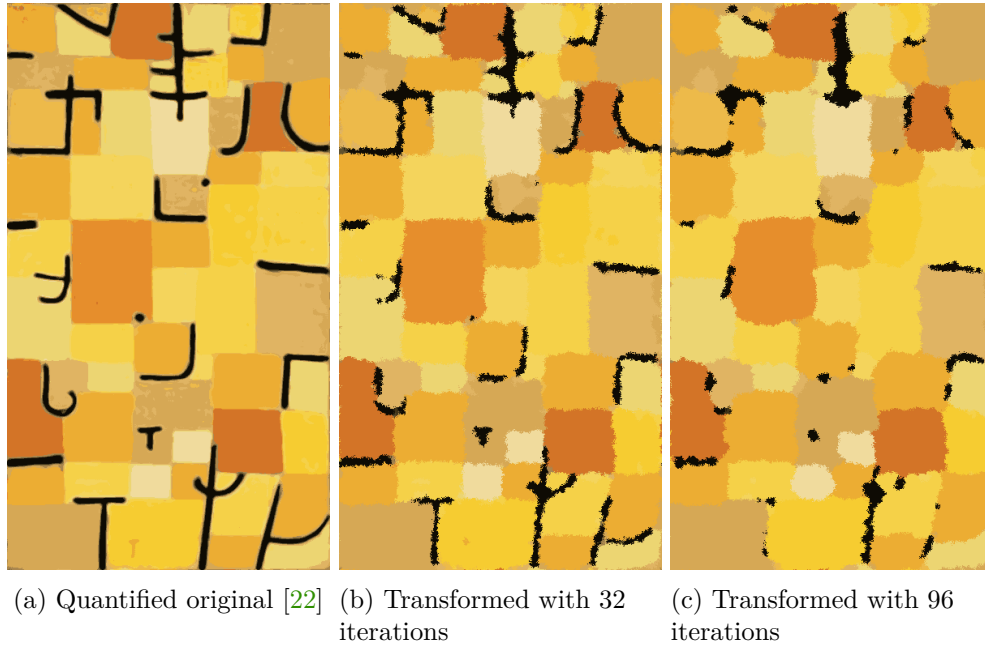


Figure 5.12: Lines dissolution effect of Median Mold rule

Neighborhood XOR The original application of this rule, as referenced in [11], is to replicate relatively small objects in black-and-white images. The rule requires a subset of neighbors, which the user can specify through the user interface. When the subset size is one, using the XOR rule is equivalent to the linear translation rule mentioned earlier. The definition of this rule is as follows:

Obtain the values of the selected neighbors. Apply bitwise XOR to all of these values. Assign the resulting value to the active cell.

Though intended as a rule for copying, this effect only holds valid for 2^n iterations and when the image is small enough to avoid self-interference. With a cellular automaton featuring more states, larger images, and no constraints on the number of iterations, the results can be more complex and intriguing, as seen in Figure 5.13.

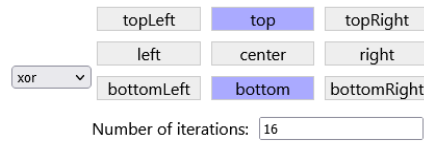
In the original paper, the cellular automaton had only two states: 0 for white and 1 for black pixels. Since RGB color representation uses an inverse mapping, with #000000 as black and #ffffff as white, it is necessary to invert the colors before displaying the final output. Fortunately, this inversion only needs to be done once at the end of the operation, rather than after each iteration.

The use of a broader set of states results in a unique effect of color blending in areas where differently colored objects overlap. This gives the rule the ability to effectively generate new colors in the image. Additionally, application of bitwise XOR can lead to the canceling out of colors. When the number of neighbors is even, colors tend to cancel each other out due to self-interference. Conversely, if the number of neighbors is odd, the original color is maintained. This phenomenon is demonstrated in Figure 5.14. Given the canceling out effect, when configured with an even number of neighbors and a low number of iterations, the XOR rule can create results visually similar to edge detection algorithms, as shown in Figure 5.15.



(a) Quantified original [7]

(b) Transformed according to 5.13c



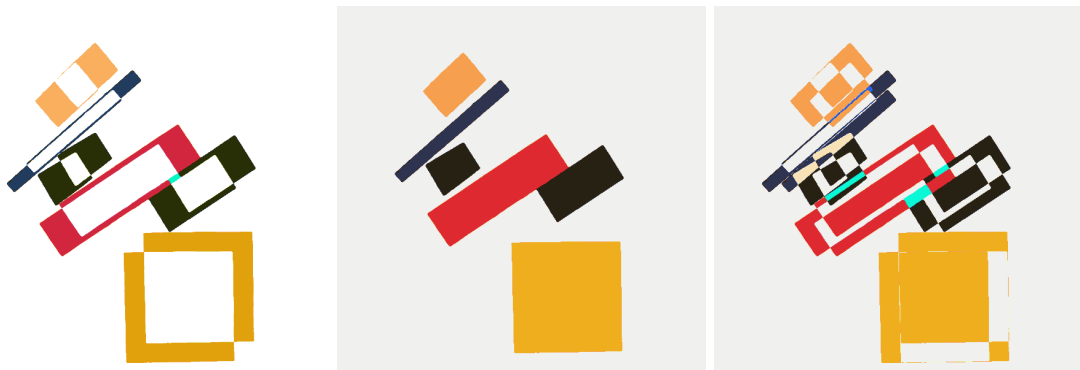
(c) User interface representation of the rule

Figure 5.13: Neighborhood XOR rule in practice



(a) XOR rule with even number of neighbors

(b) XOR rule with odd number of neighbors

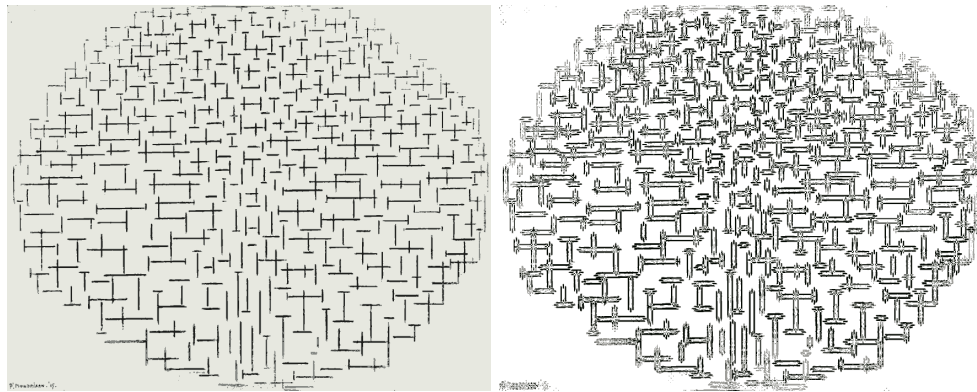


(c) Transformed according to 5.14a

(d) Quantified original [26]

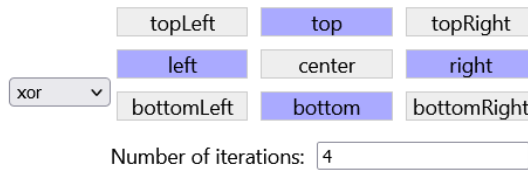
(e) Transformed according to 5.14b

Figure 5.14: Difference between even and odd numbers of neighbors for XOR rule



(a) Quantified original [30]

(b) Transformed image



(c) Corresponding setting of the used XOR rule

Figure 5.15: Effect of a XOR rule similar to edge detection algorithm

Combining the Rules Together

While the rules themselves can yield impressive outcomes, the most unexpected and creative results arise from combining them. Figure 5.16 examines one such combination, transforming a zoom artifact into a dry brush stroke effect and subsequently altering it to resemble a water surface disturbance pattern with the use of blur rule.



(a) Zoom Out & Zoom In

(b) Mold

(c) Blur

Figure 5.16: Different rules applied sequentially

The following pages showcase several improvised transformations of selected artworks, demonstrating the potential for creative reinterpretation and manipulation.

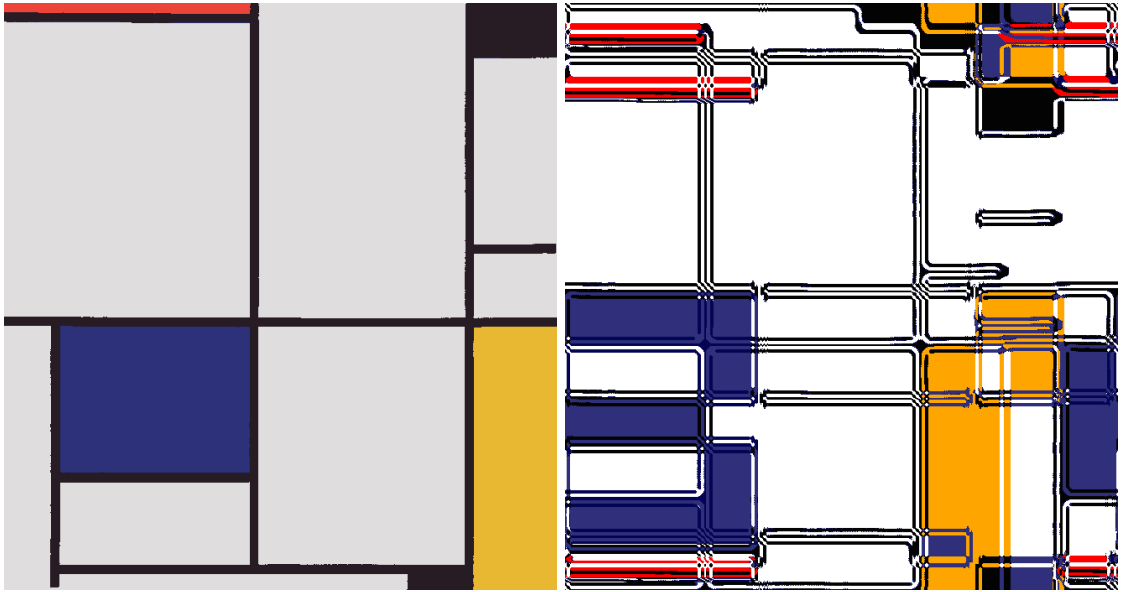


Figure 5.17: Improvisation on Mondrian [31]

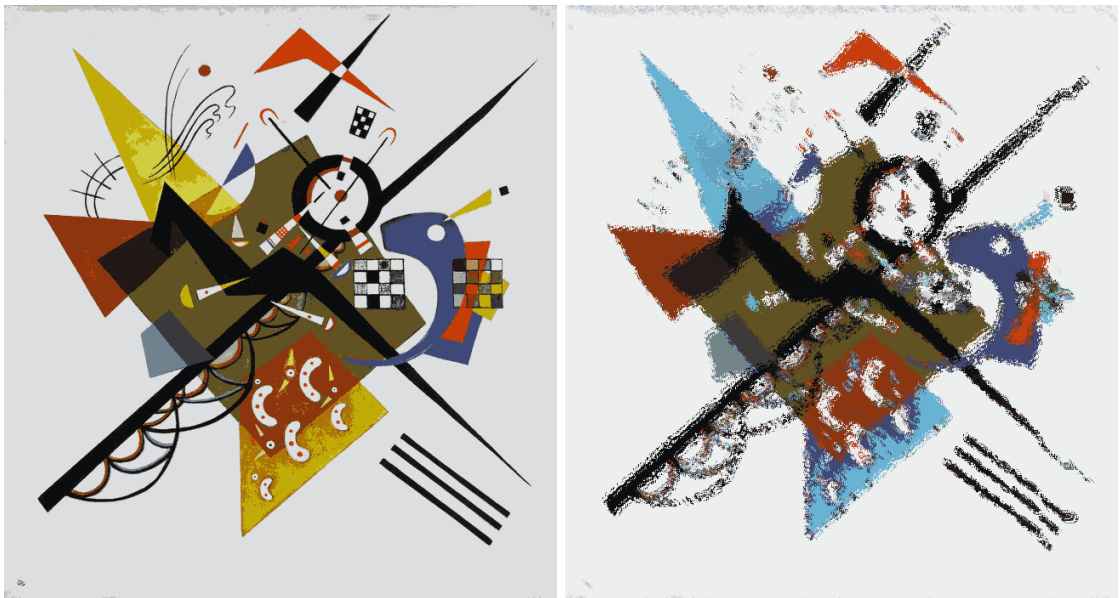


Figure 5.18: Improvisation on Kandinsky [18]

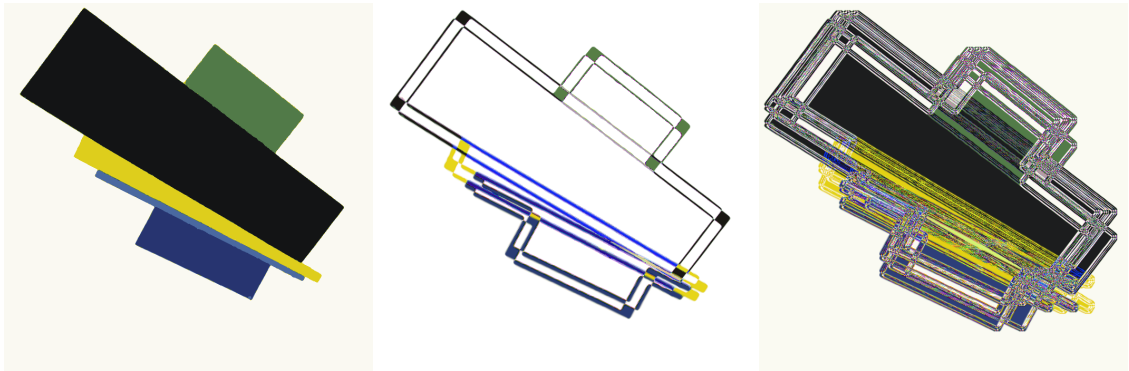


Figure 5.19: Improvisation on Malevich [27], diptych



Figure 5.20: Improvisation on Klee [20]

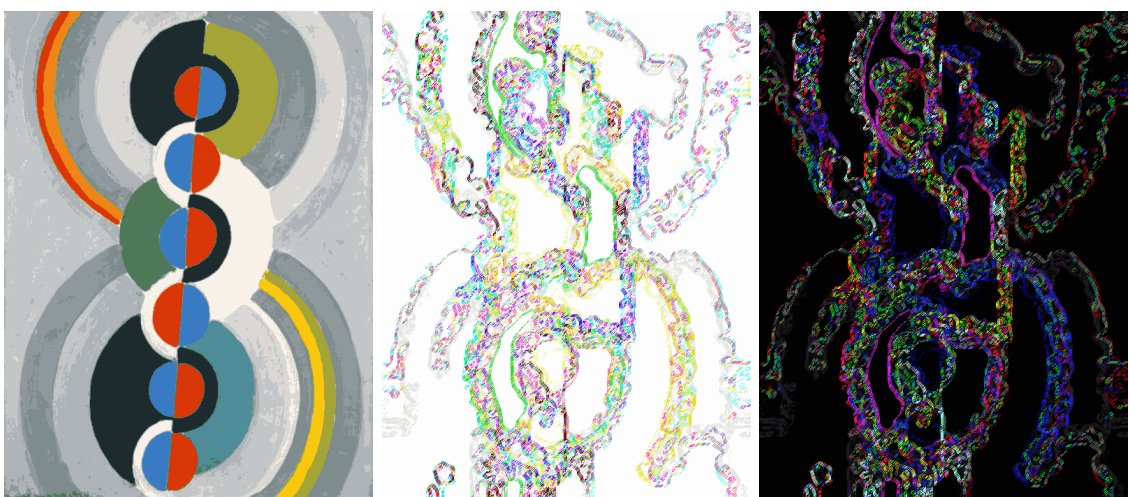


Figure 5.21: Improvisation on Delaunay [8], diptych

Chapter 6

Conclusion

This thesis presents a comprehensive exploration of the integration of cellular automaton and translation automaton in a digital art application. By combining theoretical knowledge with practical implementation, the research successfully establishes a system that enables users to experiment creatively with image transformation. The application architecture, built on the React framework, provides a foundation for handling image processing and editing through the use of various custom components. Additionally to image manipulation and exploration of unique visual transformations provided by various cellular automata rules, application features color quantization, automatic rule generation and artistic quality metrics.

The latest version of the application developed in scope of this work is available online on the following link:

https://dionysusbacchus.github.io/Art_CA/

Directions for Further Research

Despite all accomplishments, there are several areas for further development and improvement of achieved results.

In terms of performance, future work should focus on analyzing the time complexity of individual rule types and optimizing rules for larger images, more iterations, and larger palettes. Utilizing parallel processes and possibly completely separating the back end from the user interface could significantly improve efficiency. Re-implementing transformation and image processing rules to run on specialized hardware, such as GPUs, would also contribute to performance gains.

There is also room for further enhancement to elevate the user experience. Improvements could include adding loading indications, allowing users to undo rule deletions, and storing the color palette history. Additionally, testing on users and exploring different interface layouts could provide valuable insights for iterative improvements.

Moreover, adding edge detection and sub-automata capabilities could enable users to target and manipulate specific objects or segments within an art piece. Expanding the list of rules and developing new metrics, such as those for image symmetry, composition, and color compatibility, would further enhance the application's capabilities. Employing metrics and a rule generator to implement a genetic algorithm could allow users to select the number of generations, population size, and number of rules per genome. Presenting users with the top results at the end of the process would offer greater control and satisfaction.

Bibliography

- [1] ASENCIO, M. *Ml-kmeans* online. 2023. Available at: <https://www.npmjs.com/package/ml-kmeans>. [cit. 2024-02-05].
- [2] BRUCKER, J. *Paul Klee Artist Overview and Analysis* online. The Art Story Contributors, october 2011. Available at: <https://www.theartstory.org/artist/klee-paul/>. [cit. 2024-04-20].
- [3] CHOUDHURY, P. P.; NAYAK, B. K.; SAHOO, S. and RATH, S. P. Theory and Applications of Two-dimensional, Null-boundary, Nine-Neighborhood, Cellular Automata Linear rules. *CoRR*, 2008, abs/0804.2346. Available at: <http://arxiv.org/abs/0804.2346>.
- [4] CONTRIBUTORS, T. A. S. *Piet Mondrian Artist Overview and Analysis* online. The Art Story Contributors, october 2012. Available at: <https://www.theartstory.org/artist/mondrian-piet/>. [cit. 2024-04-20].
- [5] DELAUNAY, R. *Red Eiffel Tower*. 1911. Public Domain.
- [6] DELAUNAY, R. *Simultaneous Windows (2nd Motif, 1st Part)*. 1912. Public Domain.
- [7] DELAUNAY, R. *Circular Forms*. 1930. Public Domain.
- [8] DELAUNAY, R. *Rythme*. 1934. Public Domain.
- [9] FISCH, R.; GRAVNER, J. and GRIFFEATH, D. *Cyclic Cellular Automata in Two Dimensions*. 1st ed. Boston, MA: Birkhäuser Boston, 1991. 171–185 p. ISBN 978-1-4612-0451-0.
- [10] GARDNER, M. MATHEMATICAL GAMES. *Scientific American*. Scientific American, a division of Nature America, Inc., 1970, vol. 223, no. 4, p. 120–123. ISSN 00368733, 19467087. Available at: <http://www.jstor.org/stable/24927642>.
- [11] GRIFFEATH, D. and MOORE, C. *New constructions in cellular automata*. Oxford University Press, 2003. ISBN 9780195137187.
- [12] GRIFFIN, E. *Wassily Kandinsky Artist Overview and Analysis* online. The Art Story Contributors, february 2013. Available at: <https://www.theartstory.org/artist/kandinsky-wassily/>. [cit. 2024-04-20].
- [13] IMRICH BERTÓK, I. J. *Počítače & umenie*. 1st ed. Slovenské pedagogické nakladateľstvo, 1989. ISBN 80-08-00037-6.

- [14] JENKINS, S. *Robert Delaunay Artist Overview and Analysis* online. The Art Story Contributors, december 2015. Available at: <https://www.theartstory.org/artist/delaunay-robert/>. [cit. 2024-04-20].
- [15] KANDINSKY, V. *Circles in a Circle* online. Philadelphia Museum of Art, 1923. Available at: <https://www.philamuseum.org/collection/object/51019>. [cit. 2023-11-20].
- [16] KANDINSKY, V. *Composition 8* online. The Solomon R. Guggenheim Foundation, 1923. Available at: <https://www.guggenheim.org/artwork/1924>. [cit. 2023-11-20].
- [17] KANDINSKY, V. *Delicate Tension. No. 85* online. Thyssen-Bornemisza National Museum, 1923. Available at: <https://www.museothyssen.org/en/collection/artists/kandinsky-wassily/delicate-tension-no-85>. [cit. 2023-11-20].
- [18] KANDINSKY, V. *On White II* online. Obelisk Art History, 1923. Available at: <https://www.arthistoryproject.com/artists/wassily-kandinsky/on-white-ii/>. [cit. 2023-11-20].
- [19] KANDINSKY, W. *Concerning the Spiritual in Art*. Dover Publications, 1977. Dover Fine Art, History of Art. ISBN 9780486234113. Available at: <https://books.google.cz/books?id=mG-VRWgfpuYC>.
- [20] KLEE, P. *Castle and Sun* online. Totallyhistory.com, 1929. Available at: <https://totallyhistory.com/castle-and-sun/>. [cit. 2023-11-17].
- [21] KLEE, P. *Fire at Full Moon* online. Google Arts & Culture, 1933. Available at: <https://artsandculture.google.com/asset/fire-at-full-moon-paul-quee/4QFsIMp64ZsDOA>. [cit. 2023-11-17].
- [22] KLEE, P. *Characters in Yellow* online. Fondation Beyeler, 1937. Available at: https://www.fondationbeyeler.ch/sammlung/werk?tx_wmdbbasefbey_pi5%5Bartwork%5D=287&cHash=6d17d1f931685cf196547cb717c778fd. [cit. 2023-11-17].
- [23] KLEE, P. *Heroic Roses* online. WikiArt.org, 1938. Available at: <https://www.wikiart.org/en/paul-quee/heroic-roses-1938>. [cit. 2023-11-17].
- [24] MAHAMA, A. T. S.; DOSSA, A. S. and GOUTON, P. Choice of distance metrics for RGB color image analysis. *Electronic Imaging*. Society for Imaging Science and Technology, 2016, vol. 28, p. 1–4.
- [25] MALEVICH, K. *Self-Portrait in Two Dimensions* online. Artchive, 1915. Available at: <https://www.artchive.com/artwork/suprematism-self-portrait-in-two-dimensions-kasimir-malevich-1915/>. [cit. 2023-11-15].
- [26] MALEVICH, K. *Suprematism* online. 1915. Available at: https://commons.wikimedia.org/wiki/File:Suprematism_%28Malevich,_1915;_1%29.jpg. [cit. 2023-11-15].
- [27] MALEVICH, K. *Suprematism, 18th Construction* online. Sotheby's, 1915. Available at: <https://www.sothebys.com/en/auctions/ecatalogue/2015/impressionist-modern-art-evening-sale-115006/lot.18.html>. [cit. 2023-11-15].

- [28] MALEVICH, K. *Suprematist Composition (with eight red rectangles)* online. Stedelijk Museum Amsterdam, 1915. Available at: <https://www.stedelijk.nl/en/collection/3133-kazimir-malevich-suprematistische-compositie-%28met-acht-rode-rechthoeken%29>. [cit. 2023-11-15].
- [29] MEDUNA, A. *Automata and Languages: Theory and Applications*. 1st ed. Springer London, July 2000. ISBN 978-1-85233-074-3.
- [30] MONDRIAN, P. *Composition No.10 (Pier and Ocean)*. 1915. Public Domain.
- [31] MONDRIAN, P. *Tableau I*. 1921. Public Domain.
- [32] MONDRIAN, P. *Composition with Red, Yellow and Blue*. 1942. Public Domain.
- [33] MONDRIAN, P. *New York City I*. 1942. Public Domain.
- [34] SAVVINE, I. *Kazimir Malevich Artist Overview and Analysis* online. The Art Story Contributors, june 2014. Available at: <https://www.theartstory.org/artist/malevich-kasimir/>. [cit. 2024-04-20].
- [35] SHINTYAKOV, D. *The Single Rotation rule: remarkably simple and rich reversible cellular automaton* online. November 2013. Available at: <http://dmishin.blogspot.com/2013/11/the-single-rotation-rule-remarkably.html>. [cit. 2024-02-09].
- [36] SHINTYAKOV, D. *Single Rotation rule with frame interpolation* online. December 2014. Available at: <http://dmishin.blogspot.com/2014/12/single-rotation-rule-with-frame.html>. [cit. 2024-02-09].
- [37] SHINTYAKOV, D. *Now in 3D (Single Rotation)* online. January 2015. Available at: <http://dmishin.blogspot.com/2014/12/single-rotation-rule-with-frame.html>. [cit. 2024-02-09].
- [38] WIKIPEDIA CONTRIBUTORS. *History of photography — Wikipedia, The Free Encyclopedia* online. 2024. Available at: https://en.wikipedia.org/w/index.php?title=History_of_photography&oldid=1215943409. [cit. 2024-03-15].
- [39] WOLFRAM, S. *A New Kind of Science*. 1st ed. Wolfram Media, Inc., may 2005. ISBN 9781579550080.
- [40] ZAWIDZKI, M. *2D 2C R1 Cellular Automaton on a Triangular Grid* online. wolfram.com, march 2011. Available at: <https://demonstrations.wolfram.com/2D2CR1CellularAutomatonOnATriangularGrid/>. [cit. 2024-02-23].
- [41] ZAWIDZKI, M. *Hexagonal Outer Totalistic Cellular Automata* online. wolfram.com, march 2011. Available at: <https://demonstrations.wolfram.com/HexagonalOuterTotalisticCellularAutomata/>. [cit. 2024-02-23].

Appendix A

Contents of the included storage media

The attached medium contains the following items:

- `README.md` An equivalent note with the list of stored items and instructions for installation
- `sources/` Folder containing source files of the application implementation
- `thesis/` Folder containing source files of the thesis text
- `xkhrys00_thesis.pdf` Compiled PDF version of this thesis

Running Application

Online Deploy

Latest version of the application is available online on the following link:

https://dionysusbacchus.github.io/Art_CA/

Installation Instructions

Running application on the local host requires NodeJS of version v20.12.2 or higher. To start local server, run following commands in the `sources` folder:

```
npm install
set NODE_OPTIONS=--openssl-legacy-provider
npm start
```