Code Painting: A Computer Scientist’s Story of AI and Art

Original Source: **Daejeon Biennale 2020** — Artificial Intelligence: Sunshine Misses Windows
Author: **Joo-Haeng Lee**, PhD (Co-founder/CEO of Pebblous, Principal Researcher at ETRI, Professor at UST, Artist)

Introduction

While preparing my artwork for the Daejeon Biennale, I was asked to write an essay on "Artificial Intelligence and Art." As a computer scientist researching AI at a research institute, and as an artist who uses it to create works in my spare time, I thought this was a good opportunity to share my experiences and thoughts on AI and Art.

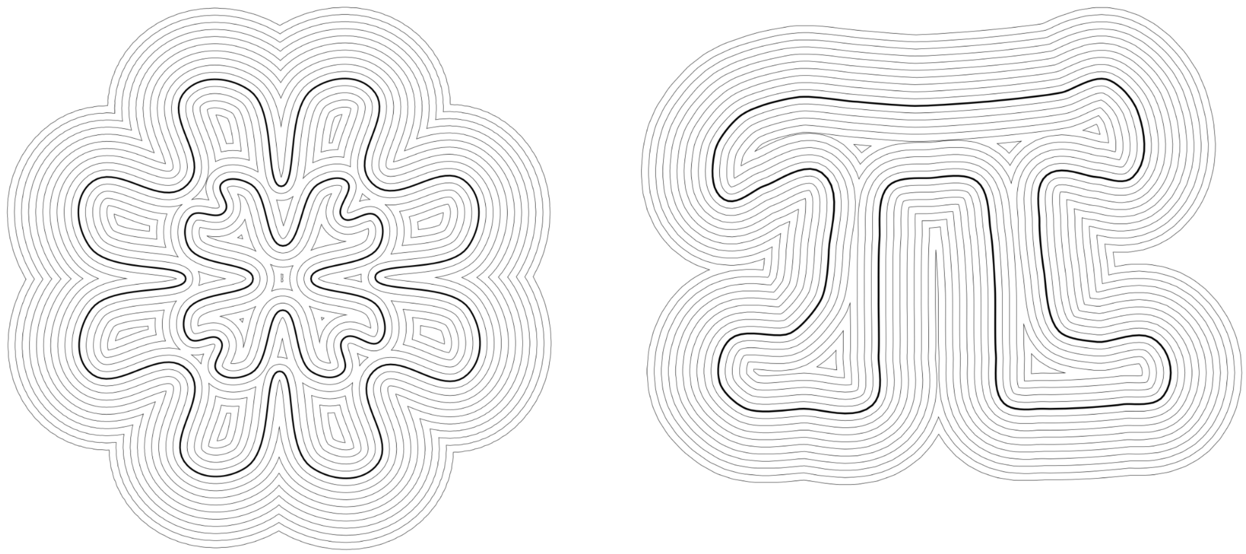
In the first part of this essay, I will start lightly with my personal story of how I entered the world of computer painting and my debut as an artist last year—a time that was both exciting and full of trial and error. In the latter half, I plan to explain the meeting of science and art, specifically the meaning that recent developments in AI technology hold for art, based on my personal experience.

As an active researcher at Daedeok Innopolis and an artist who gained a precious debut opportunity through the Daejeon Science Festival, I prepared this writing to support and cheer on the new direction of art presented by the 'Daejeon Biennale'.

Entering the World of Computer Painting

I studied Computer Science in undergraduate and graduate school. My specific major in graduate school was Computer Graphics. Unlike the computer graphics taught in art schools, the computer graphics learned in engineering schools is based on mathematics and physics. We create mathematical and physical models to express shape, movement, and color, and simulate them on a computer.

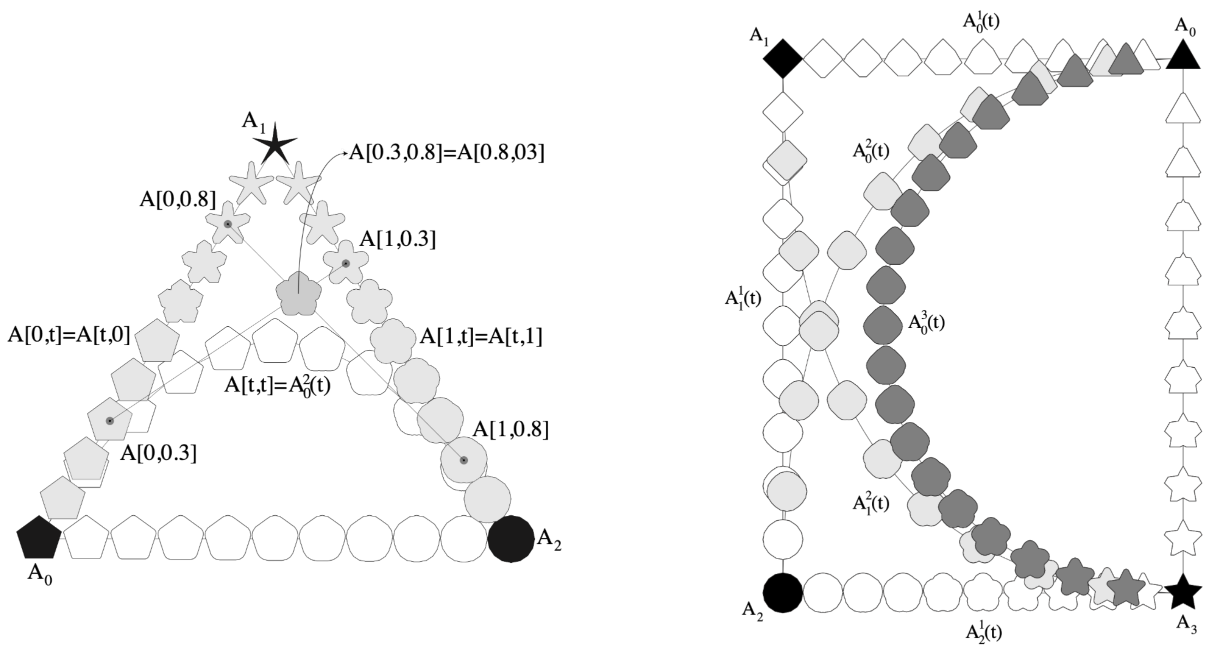
My connection with computer graphics actually began with computer music. While conducting small computer music experiments in my university dormitory, I participated in making background music for an animation project in the graduate computer graphics lab. I was fascinated not only by the excellent research equipment and the warmth of the seniors but also by the fact that I was participating in the work of creating animation with a computer. Eventually, after participating in undergraduate research, I decided to enter graduate school in the Computer Graphics Laboratory.



[Figure 1]
Description: A PhD research study on the offset curves of free-form curves. It expresses the trajectory of circles of different sizes moving along a curve. While it has significant meaning in industrial settings, the form itself is attractive. (1999, POSTECH)

However, once I joined the lab and began research in earnest, the proportion of animation work decreased. Instead, I was able to research 'pictures' properly. Although I never had the chance to learn painting properly during my school days, I personally enjoyed drawing and studied art history quite happily. Therefore, studying how to draw pictures with advanced computers was incredibly exciting.

However, serious computer graphics research is somewhat distant from 'art.' While aesthetic use of research results is considered, the important thing is to master deep mathematical theories and difficult computer algorithms. My advisor majored in mathematics and double-majored in computer science, so his theoretical depth in computer graphics was immense. Thanks to studying with him, I was able to experience a fairly high-level world of mathematics, unlike ordinary computer science graduate students. Although a long time has passed since graduation, the experience of mathematics and abstract thinking I studied then has been a great help in my subsequent engineering research and artistic activities.



[Figure 2]
Description: Research on Shape Blending using the Direction Map of polygons (2003).

No matter how armed with theory one is, the results of computer graphics research must be accompanied by cool pictures along with beautiful formulas. Creating beautiful pictures with existing theories and challenging oneself to find new theories to create better pictures was a very attractive process. Also, being able to draw cool pictures at will using the best computer graphics equipment at the time was a sweet reward for studying difficult theories. Indeed, the computer graphics major was a very ideal field for me, where research and hobbies became one.

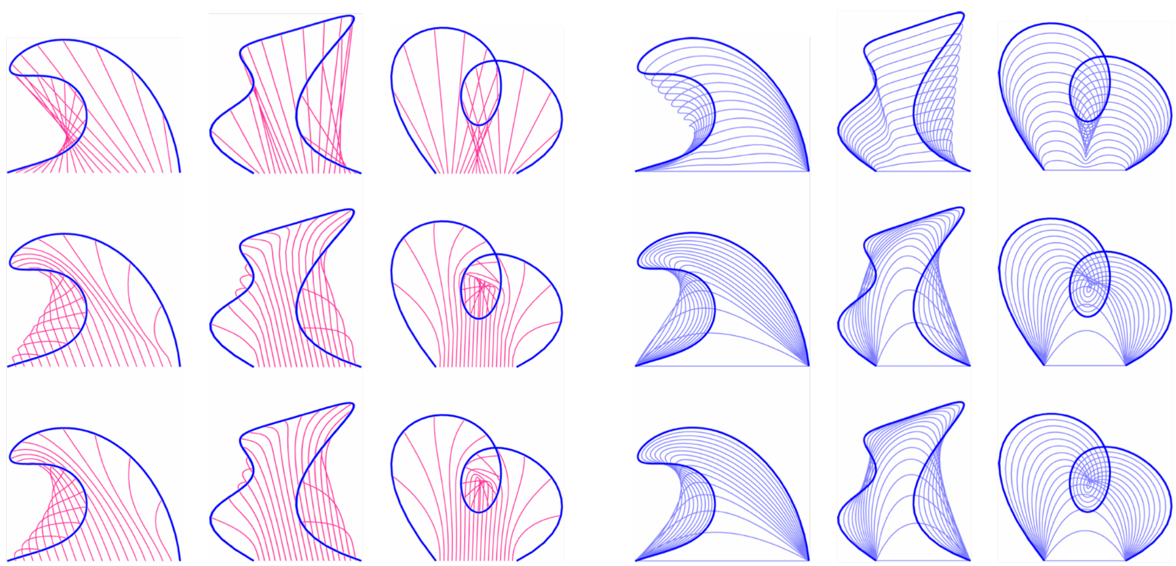
Looking back now, what I mainly learned in school was "how to observe reality, model it, and simulate it with a computer." When the simulation results were good, they were expressed as beautiful images, so I could enjoy visual luxury as a bonus. However, ultimately, what was important were the process of abstract thinking and the tools of mathematics and physics needed to model complex reality. These experiences in modeling and simulation, along with knowledge of mathematics and physics, became the foundation for researching various fields beyond my graduate major while working at a research institute after obtaining my degree.

Drawing Imagination with Code: What and How to

Draw?

I joined a research institute as part of my military service immediately after graduating. In the early days, I spent busy and enjoyable days. Above all, I was able to try various new experiments based on the knowledge I studied in graduate school. In particular, the research team I worked with at the time implicitly guaranteed very smooth communication and free research, unlike the usual atmosphere of government-funded research institutes. Thanks to this, I was able to write papers dealing with more fundamental research topics in addition to the projects I performed in exchange for my salary.

From the early 2000s to the mid-2010s, the basis of my research was modeling and simulation based on mathematical and physical knowledge. From robots to rendering technologies, although the research topics were different, the methodology was the same. And for research presentations, I always used very carefully prepared pictures. I also put a lot of effort into creating presentation materials. This was partly due to the training I received in graduate school, but it was also my personal tendency to pursue completeness in both content and form.

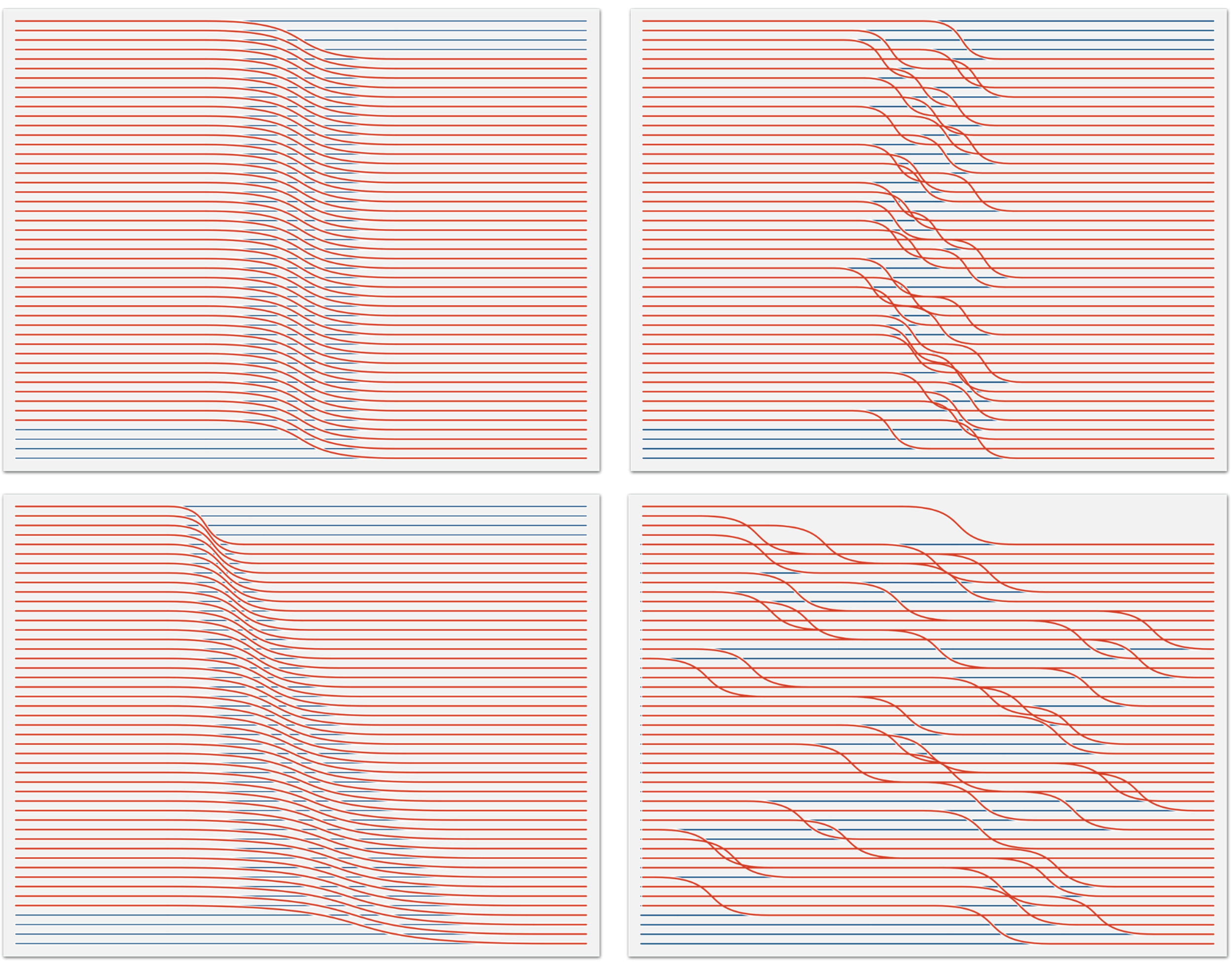


[Figure 3]
Description: Growth structure of high-order Bézier curves (Rib and Fan). A geometry study starting from the imagination: "Does every curve have a unique fingerprint?" (2006)

I'm not sure when it started, but I began to think of thesis presentations as a "short play" or "show" well-prepared for the audience. Although the figures in the thesis, the presentation materials, and the presentation itself are all means of objective knowledge communication, I think my aesthetic tastes were reflected. This background seems to have been a great help in my long practice and recent artistic activities.

The tools for drawing with a computer are very diverse. Tools chosen vary depending on the case. Although image tools like Photoshop are sometimes used in the final stage, pictures by engineers who majored in computer graphics are mostly created with 'code.' Code is also called a 'computer program.' Just as people use language for communication, we need a special language when we give work to a computer. This is called a programming language.

Code is a special 'writing' written in a programming language. Sometimes there is beautiful code, but mostly it is very long, difficult, and boring explanatory text. However, through this code, human thoughts can be conveyed to a computer. Thanks to the development of AI technology, modern computers give answers readily even to simple inputs, but in most cases, you still have to give very detailed instructions to get the desired answer.



[Figure 4]
Description: Example of expressing changes in the mind using geometric symbols. Connected Lines 4 Streams (2017).

Thus, according to the code entered by a human, the computer performs the task and outputs the result. There may be outputs of text, sound, and movement, but computer graphics mainly deals with image output created by the computer. Looking at the output image, one can sometimes see errors in the code. In engineering research, drawing usually has a predetermined purpose. If there is an element in the picture that goes against this purpose, it is considered an error, and the code creating the picture is modified.

The process of writing code and checking the picture is usually repeated many times. This process usually takes a very long time and is quite boring. Therefore, coding can be seen as a kind of knowledge labor process. Nevertheless, if good results can be obtained, it can be sufficient compensation for the labor.

But is coding that is enjoyable in itself impossible? Can't we create beautiful pictures with reduced labor? Can't it be immediate and intuitive, like a painter working on a canvas with a brush dipped in paint?

This is greatly influenced by which programming language is used. When building large systems that actually operate, strict programming languages like C or Python are used. However, if you use these programming languages when drawing creative pictures, the inspiration to draw may be broken. The preparation is complex, and interactive feedback is weak.

Conversely, there are programming languages like the Wolfram Language. The Wolfram Language runs on special software called Mathematica. This language was created to allow scientists to focus on problem-solving, and its great advantage is that intuitive coding is possible. In particular, you can quickly sketch and experiment with ideas that come to mind without complex preparation for the program. It's like drawing an idea on a napkin while drinking coffee. It is excellent for modeling and simulation, the basic methods of computer-based science, but it is also very useful just for drawing pictures. Even philosophers and writers use it to organize and convey their ideas.

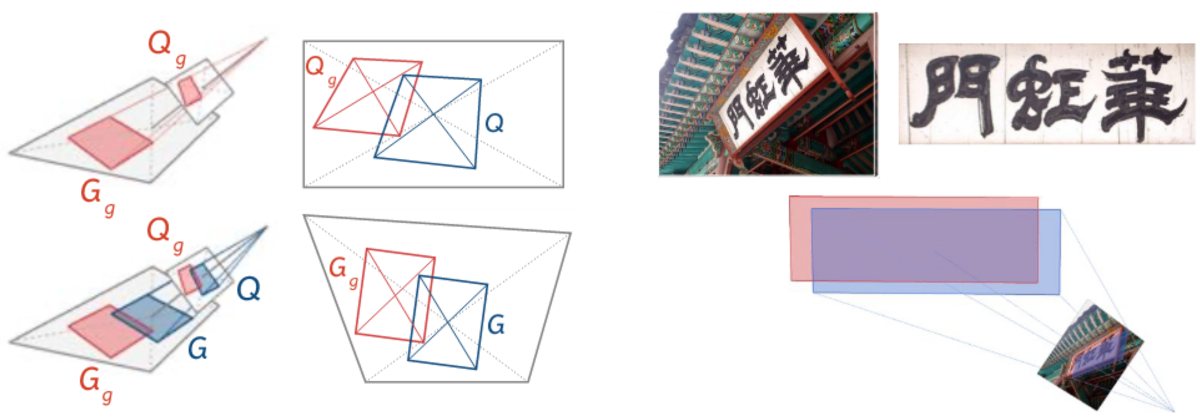
I have been using Mathematica since graduate school. It was mostly for drawing pictures with the purpose of knowledge transfer. However, perhaps because of the convenience of the tool, I started using it for 'whimsical' work. This is the beginning of 'Code Painting.' In addition to the pictures drawn for knowledge transfer, I also drew pictures containing imagination and heart. The two types of pictures were different in content, but their production method and format were identical.

Don't you sometimes doodle in the corner of a notebook while studying? It was exactly that process. More elegantly, some people compared it to Literati painting (Muninhwa). Did not the old scholars draw the world of ideas or record play with the brush, ink, and paper used for studying? Since I drew pictures with the coding I used for research, the context is certainly connected.

Meeting AI - Discovery of a New Fire

Visual expression utilizing theories of mathematics and physics, and computer models and simulation methods, was my most important research topic until around 2015. It was a time when I pursued expressing any phenomenon with concise formulas and symbols. For example, there was a case of a geometric approach to the principle of a camera. Although it is an old research field, I found a new question and tried various things to find the answer. More specifically, it was a problem of finding the exact shape of a rectangle and the position and rotation of the camera simultaneously from a single camera image containing a rectangle whose aspect ratio is unknown.

I described the vague question in my head according to mathematical forms and went through a process of numerous visualizations and formula refinements to find the answer. The formula finally found was concise and beautiful.



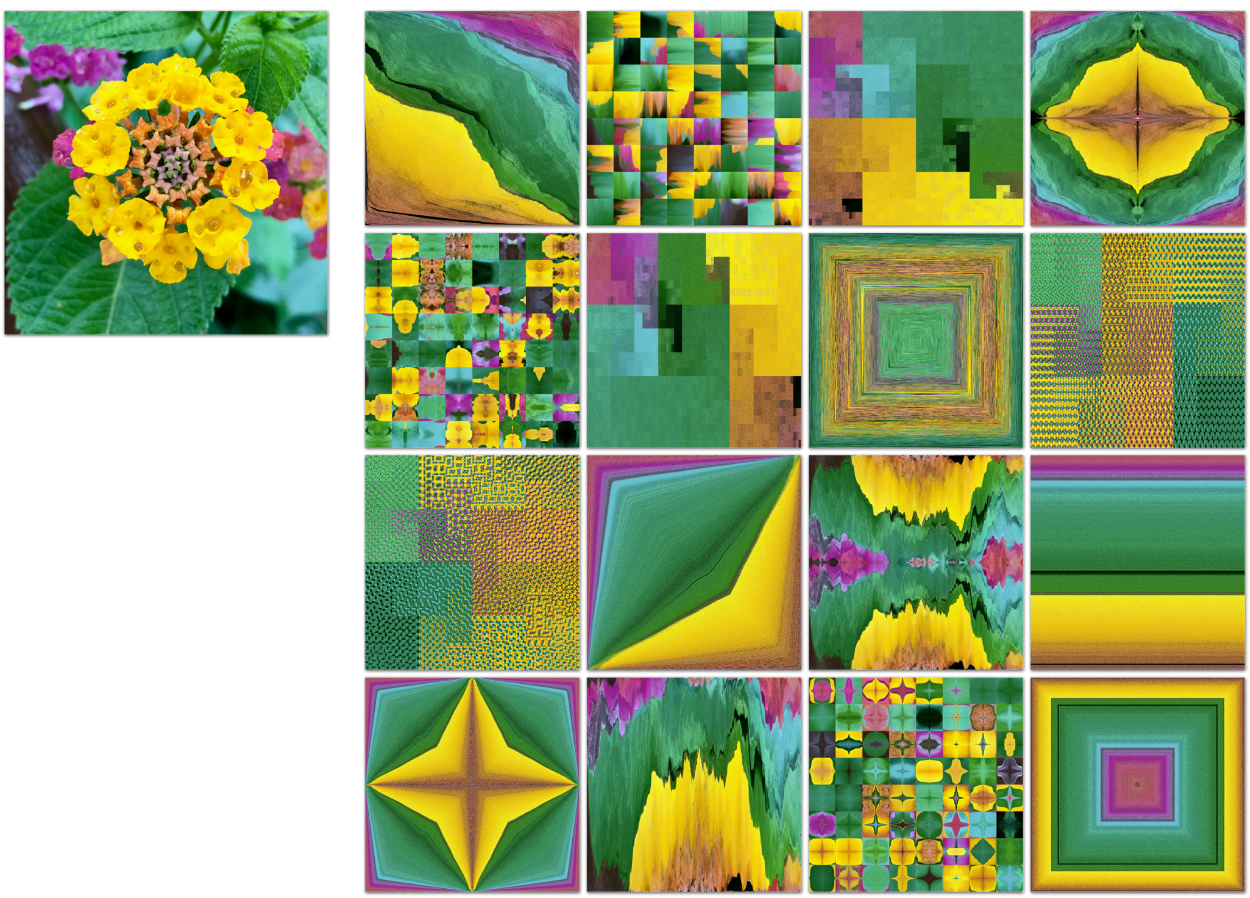
[Figure 5]
Description: Geometric research on the relationship between a rectangle and a camera. From a single photo of a rectangle whose shape is not exactly known, the exact shape of the rectangle and the shooting position of the camera can be known at once. (2012, 2013)

However, recently, a new field of interest has emerged that deviates from this framework. A strange species I had never seen before appeared, and I couldn't take my eyes off this guy. So I kept looking at it and fell for its fatal charm. However, observing this guy closely, who I thought was perfect, revealed that he was full of flaws. Many people say that this is at the center of the '4th Industrial Revolution' and that countless jobs will be threatened because of it. It is Artificial Intelligence (AI).

So what charm of AI attracted my attention? Even though it is vastly different from the mathematical methods I previously liked. It seems it is because I experienced a new dimension of coding thanks to AI.

If the existing dimension was for a human with high-level knowledge and sufficient training to find the answer to a limited range of problems through the sophisticated interaction of symbols, the new dimension is that anyone can find the answer to a huge problem through a combination of unstructured patterns found in sufficient data by a high-performance computer. Although we still don't know exactly how this new dimensional approach works, and its operation is not perfect.

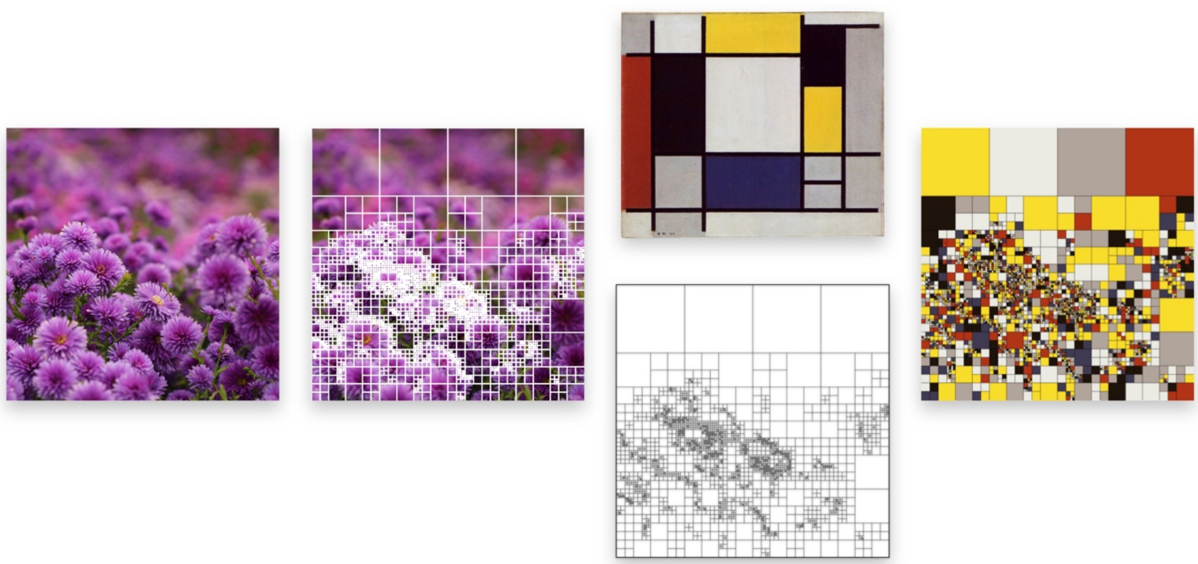
Observing complex phenomena, finding concise formulas to explain them, and hand-coding them to create beautiful pictures is an elegant and intellectual process. Although manual work is required to create code, there is pride that this is also a field that requires the touch of a coding craftsman stitch by stitch. To someone who loved this way of working, the recent AI method was quite surprising, unfamiliar, and embarrassing. Is it like a situation where a handsome rich kid who transferred from Seoul ranked first on the first exam, but sounds a bit like a fool when he talks? I want to share this feeling, so let me explain a bit more.



[Figure 6]

Description: Example of patterns created by reconstructing symbols, Lantana and 4x4 Pixel Stack (2018).

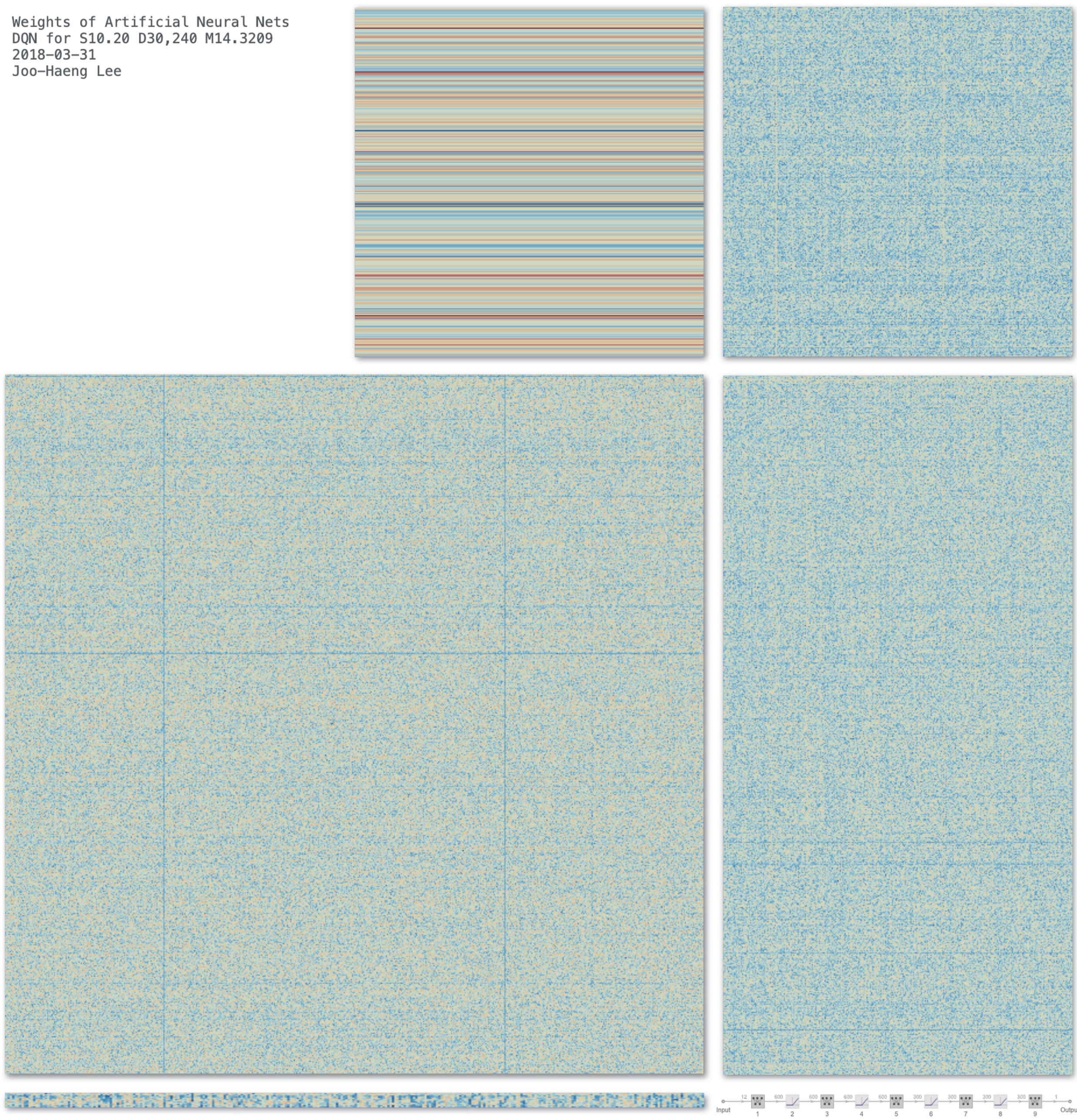
Let's think about the human role in 'Code Painting.' Code is executed by a computer, but creating that code is the human's role. First, the work the computer will do must be defined. First, definitions for input and output are needed. Next, the work procedure to create the output from the given input must be defined. The computer does not know the procedure at all. Instead, the human must know it exactly. Once the input, output, and procedure are defined, a process of transferring this to code as is is required. Unlike when giving work to a smart person, when giving work to a computer, you have to explain it 'kindly' to the point of being excessive. This is because it operates exactly as written in the code. Therefore, coding is a field that requires high proficiency and a lot of time and effort.



[Figure 7]
Description: Experiment automatically generating Mondrian-style compositions from photos. (Birth of Abstraction, 2018)

In this context, the amazing thing about AI is that even if only the input and output of the task are defined and the procedure is not defined, it does the work itself. If we don't call this 'intelligence,' what else would be intelligence? For example, classifying a million different photos into a thousand categories is a task where input/output definition is sufficiently possible. However, actually specifying the detailed procedures required for classification is almost impossible for humans. Therefore, writing the corresponding code is also impossible.

However, if AI is given a task definition and data that conforms to it, it can create the code that performs according to the task definition on its own. Even though humans haven't explained the procedure in detail! However, the amount of required data must be enormously large, and a special machine learning method and computing device are needed to create the code.



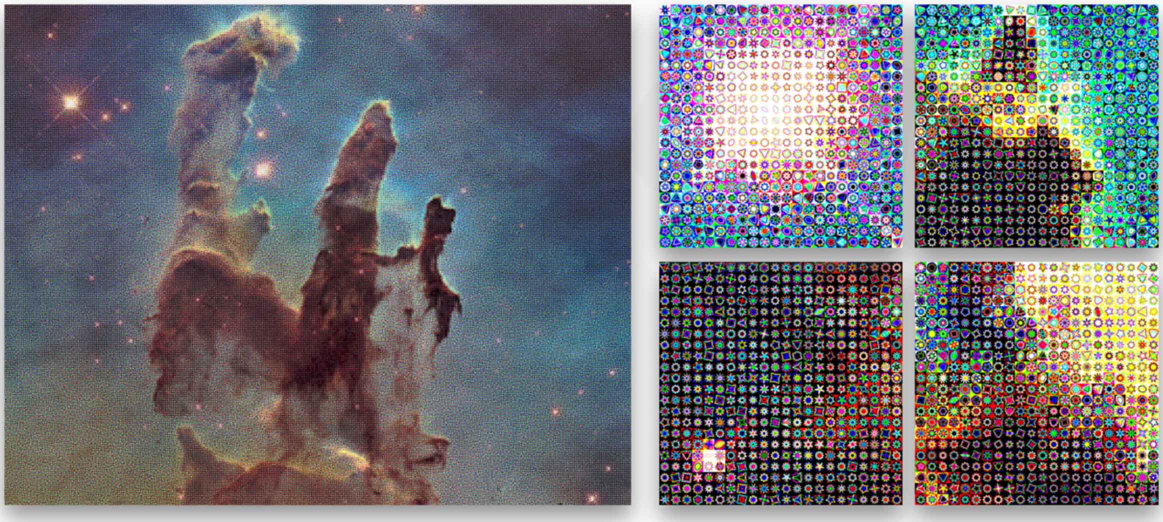
[Figure 8]
Description: Example of visualization of Deep Reinforcement Learning Neural Network. Case of Streams Game. (2018)

The method where the computer creates the code itself without human intervention from given data has been a long-standing topic in AI. Among various methods, recent attempts are directly related to Artificial Neural Networks (ANN). I first encountered ANN in graduate school in the 1990s, but theoretically, I didn't feel elegance, and practically, its operation was limited to very small problems at the time. So ANN was a research topic completely out of my interest.

Not only me, but many researchers at the time turned away from ANN. However, a small number of overseas researchers steadily developed ANN techniques, and today it has developed into a technique called Deep Neural Networks (DNN). Conceptually, DNN is not significantly different from existing ANN, but there has been a tremendous increase in the complexity and scale of the neural network. Therefore, the amount and types of data that can be handled have also greatly increased. Also, the development of special computing devices like GPUs needed to drive neural networks accompanied this. In other words, algorithms, data, and hardware—this situation where the three beats matched well—was very favorable for the development of DNN. Thanks to this, we became able to solve huge problems that we couldn't even dare to attempt with existing methods. Finally, the era of code created by computers themselves has arrived.

Go (Baduk) was one of the huge problems solved by AI in this way. So AlphaGo's victory became a shocking and symbolic event that announced the presence of DNN-based AI techniques to the world. Thanks to this, computer scientists in various fields, including computer graphics which was very cold towards AI, began to seriously accept the data-based coding method called DNN. Various academic fields such as physics, neuroscience, medicine, and economics also began to seriously consider the use of the new tool called AI and actively accept it.

I was also able to take off my colored glasses regarding AI around this time. However, rather than discarding existing elegant mathematical methods, I came to accept data-based automatic coding methods as one of the new tools.



[Figure 9]
Description: Drawing using a star-shaped virtual training set, Star-MNIST, created for AI experiments. (Star Swap - Pillars of Creation Nebula, 2019)

However, I am still cautious about current AI. There is plenty of homework to solve, such as the fact that the amount of data required for learning must be excessively vast, the energy consumption related to carbon emissions is excessively large, and that humans cannot understand its internal operating principles. Nevertheless, the achievements of current AI are also clear. We have become able to handle huge scales of data and complex problems that were undreamt of in the past history of human civilization.

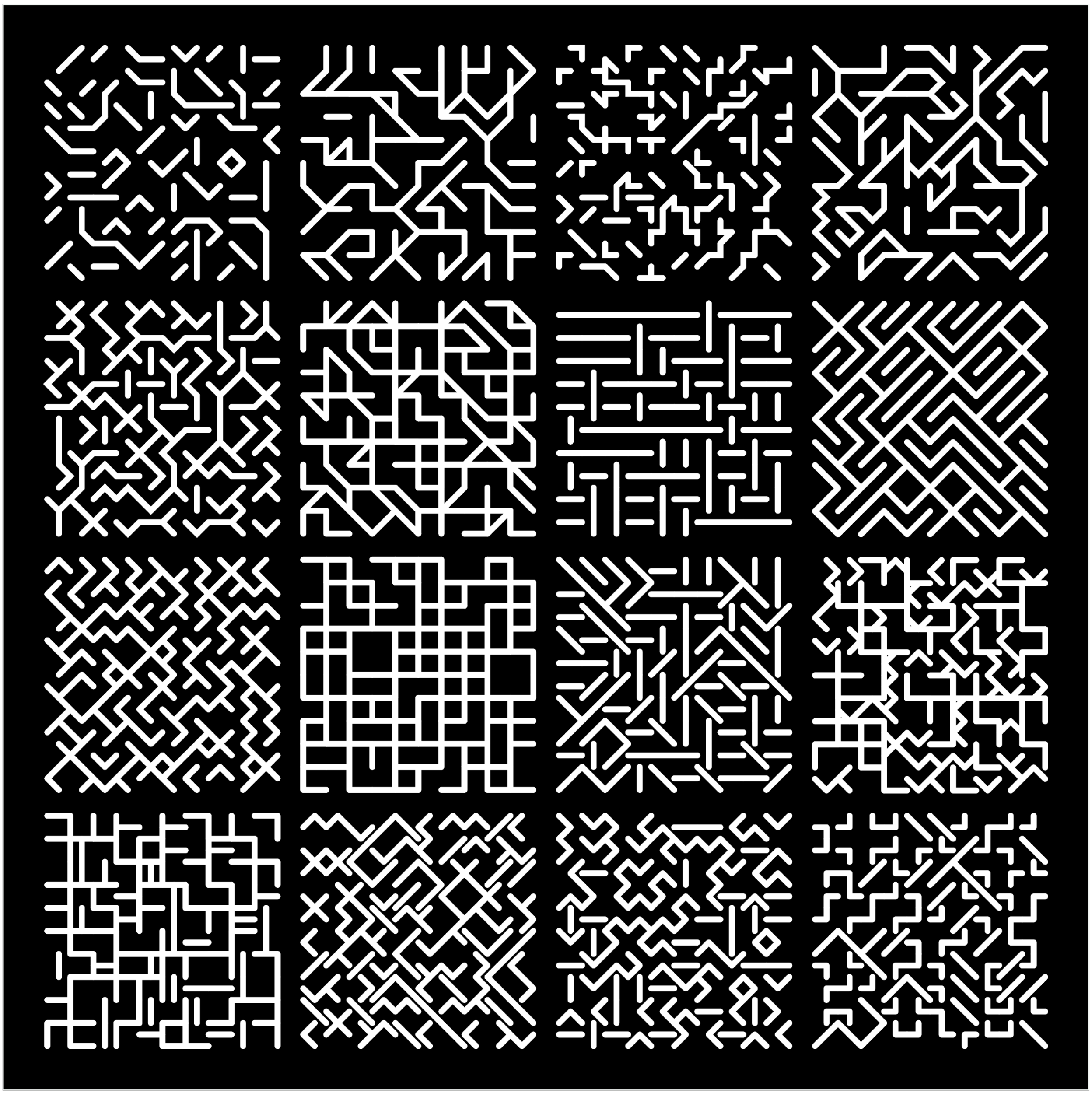
I think this is similar to when humanity first discovered fire. We first learned that meat roasted on fire tastes amazing, and everyone falls for the splendid charm of fire and performs various experiments. However, because we haven't conquered fire yet, we sometimes let the fire go out or get burned by it. But humans eventually conquered fire and went through several industrial revolutions to achieve today's civilization. Then, can humans identify the principles of AI, conquer it, and use it as a stepping stone to advance to the next civilization?

Enjoying Errors - Expanding the Horizon of Senses

With this question in the background, I would like to explain the influence AI has had on my 'Code Painting' work. Current AI is challenging many functions that existing computer graphics technology could not provide, and in some areas, it has already surpassed existing techniques. Among various possibilities, I like the fact that complex patterns can be easily created using AI the most.

However, I do not insist only on new tools. If pictures drawn with existing mathematical methods are watercolors, pictures drawn with AI code can be seen as oil paintings. Although they are completely different materials, they are common in that they are tools for painting, and I can choose differently or mix them according to need.

And more importantly, AI is an innovative tool that opens up possibilities for new expressions and gives creative inspiration, beyond just being a unique tool for drawing. For example, the 'Line Grids' series exhibited at this Biennale contains various expression methods, from symbols using mathematical tools to patterns using AI. In particular, the work based on complex patterns corresponding to the latter part of this series would have been impossible without AI methods. Thanks to this, I was able to find a new method for generating unstructured patterns, and through the works created this way, it became an opportunity to rethink the relationship between structured symbols and unstructured patterns.

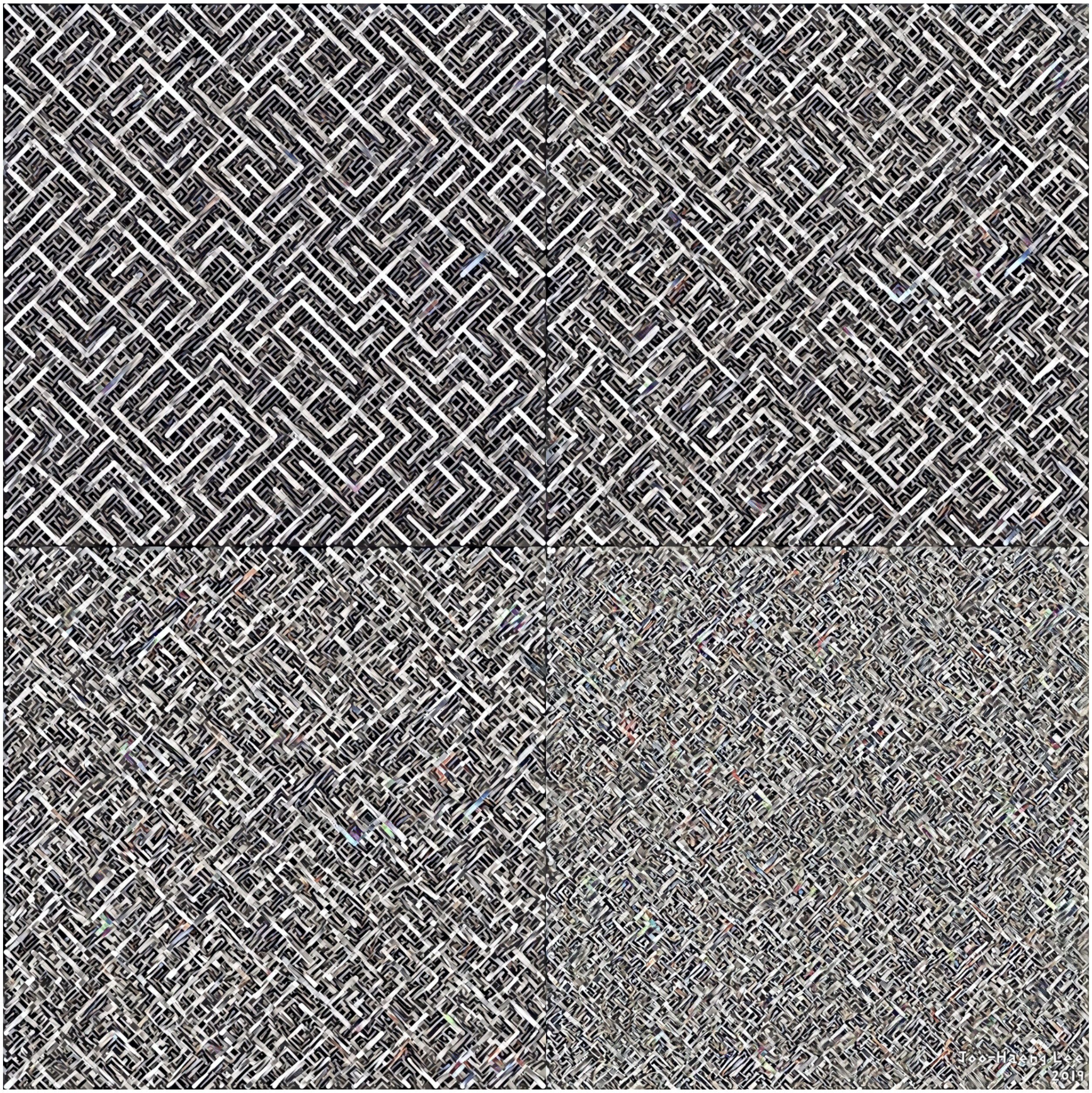


[Figure 10]
Description: Example of geometric patterns created by repetition of line segments. Atlas of Line Grids - 16 Tribes (2018).

Atlas of Line Grids - 16 Tribes is a picture expressing abstract imagination and concepts with geometric symbols. It is the result of an experiment starting from the question "Can a simple line segment be a gene for a complex pattern?" and creating new patterns through the repetition and overlapping of line segments. This way of working is the method I enjoyed before meeting AI. However, from the production of *Line Grids - Evolution of Disorder*, I started utilizing AI techniques. I created complex patterns by additionally using AI techniques on the underpainting created using existing geometric symbols and used them for artwork production.

It is different from the method of leaving everything in artwork production to the code created by AI. Let's look at this method a little more below.

Among AI techniques dealing with images, there is a method called 'Style Transfer.' Usually, Style Transfer requires two input images. One is a Style Image containing the form of drawing. The other is a Content Image containing the content of the picture. For example, if you put an image of Van Gogh as the style and a flower photo as the content, the Style Transfer Deep Neural Network modifies the flower photo by imitating Van Gogh's painting style. It creates a flower that looks as if Van Gogh painted it.



[Figure 11]
Description: Example of pattern generation using Style Transfer. It captures the 'process of collapsing' a geometrically defined structured pattern step-by-step through Deep Learning. Line Grid - Evolution of Disorder (2018).

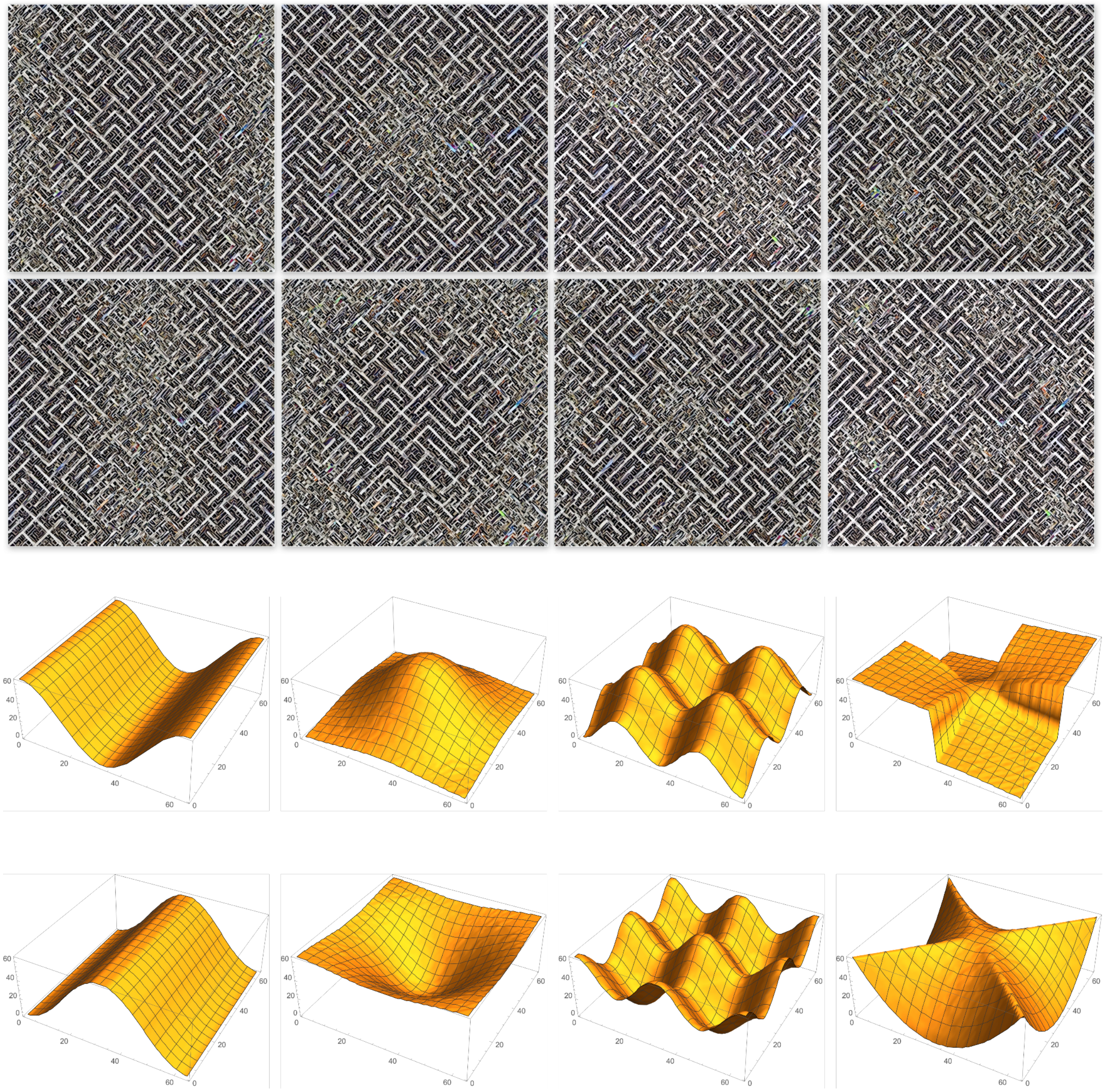
I tried an interesting experiment here. I put the same Line Grid pattern image as both the Style and Content inputs. If a human had to calculate style transfer in this situation, they would have easily recognized that the two input pictures were the same and would have outputted one of the input images as is.

However, the AI Style Transfer did not recognize that the style and content were identical. Instead, it started transforming the structured pattern received as input into an irregular pattern. Furthermore, the degree of irregularity seemed controllable.

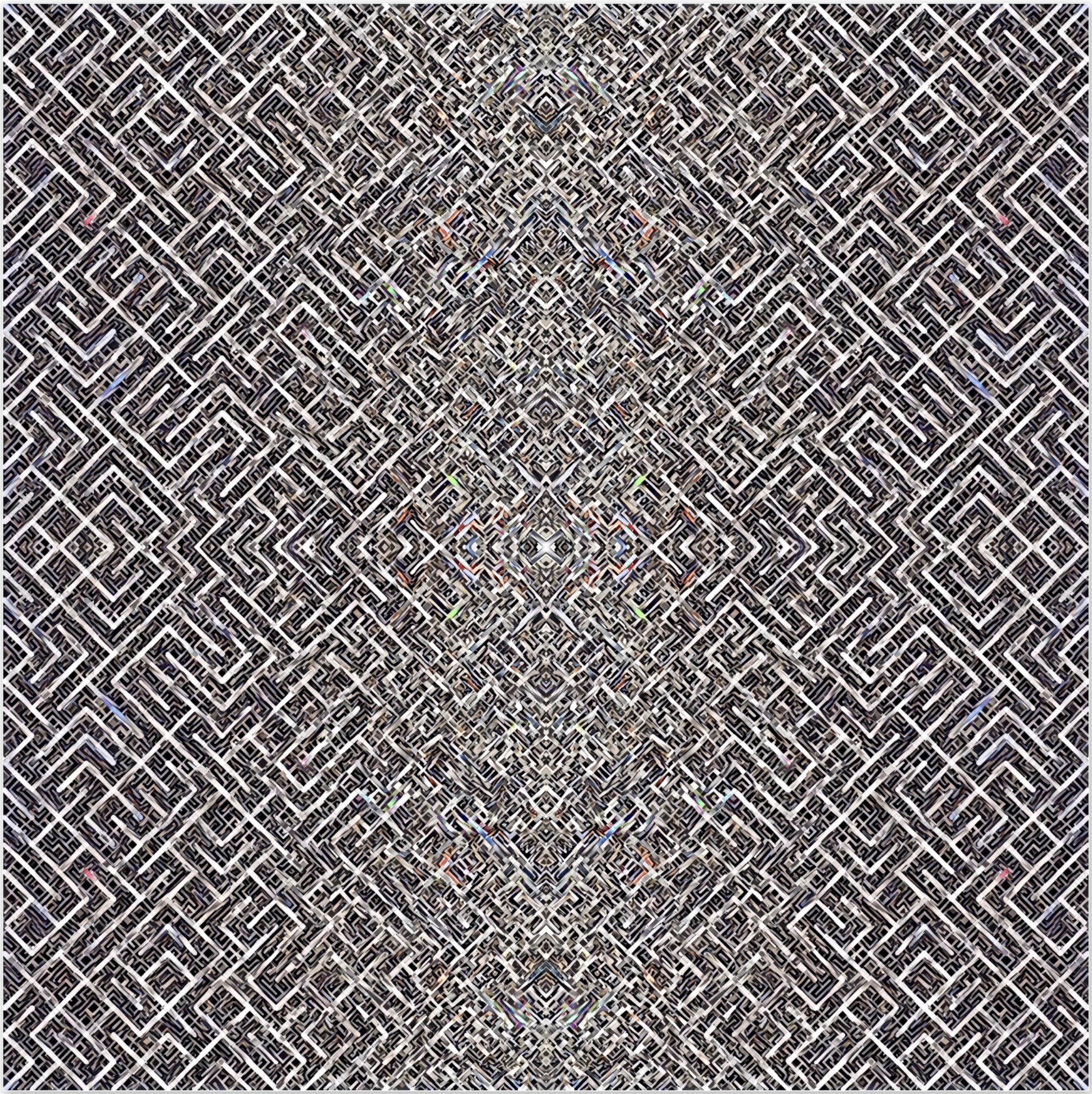
This was an interesting discovery. I expected a pattern similar to the input but somewhat different would be created, but I couldn't imagine its concrete appearance in my head. Consequently, through this experiment, I found a very simple method to create unstructured patterns from structured geometric symbols. It was an unexpected harvest.

It was impossible with existing mathematical methods alone, and it was the result of adding AI methods and utilizing them differently. Applying engineering standards, it can be seen as discovering an error in Style Transfer that occurs when style and content inputs are identical. Thanks to this, I was able to start new research to understand and eliminate the error. Furthermore, artistically, I discovered a new expression method starting from an error.

It was a welcome error and a joyful error. The interpretation of such errors was a new world from an engineering perspective. To utilize errors for artwork production in earnest, reproduction of errors had to be possible at will. Eventually, a process of 'Error Conquest' was necessary.



[Figure 12]
Description: Line Grid - (a) Evolution of Disorder, and (b) the mathematical surfaces that generated it (2019).



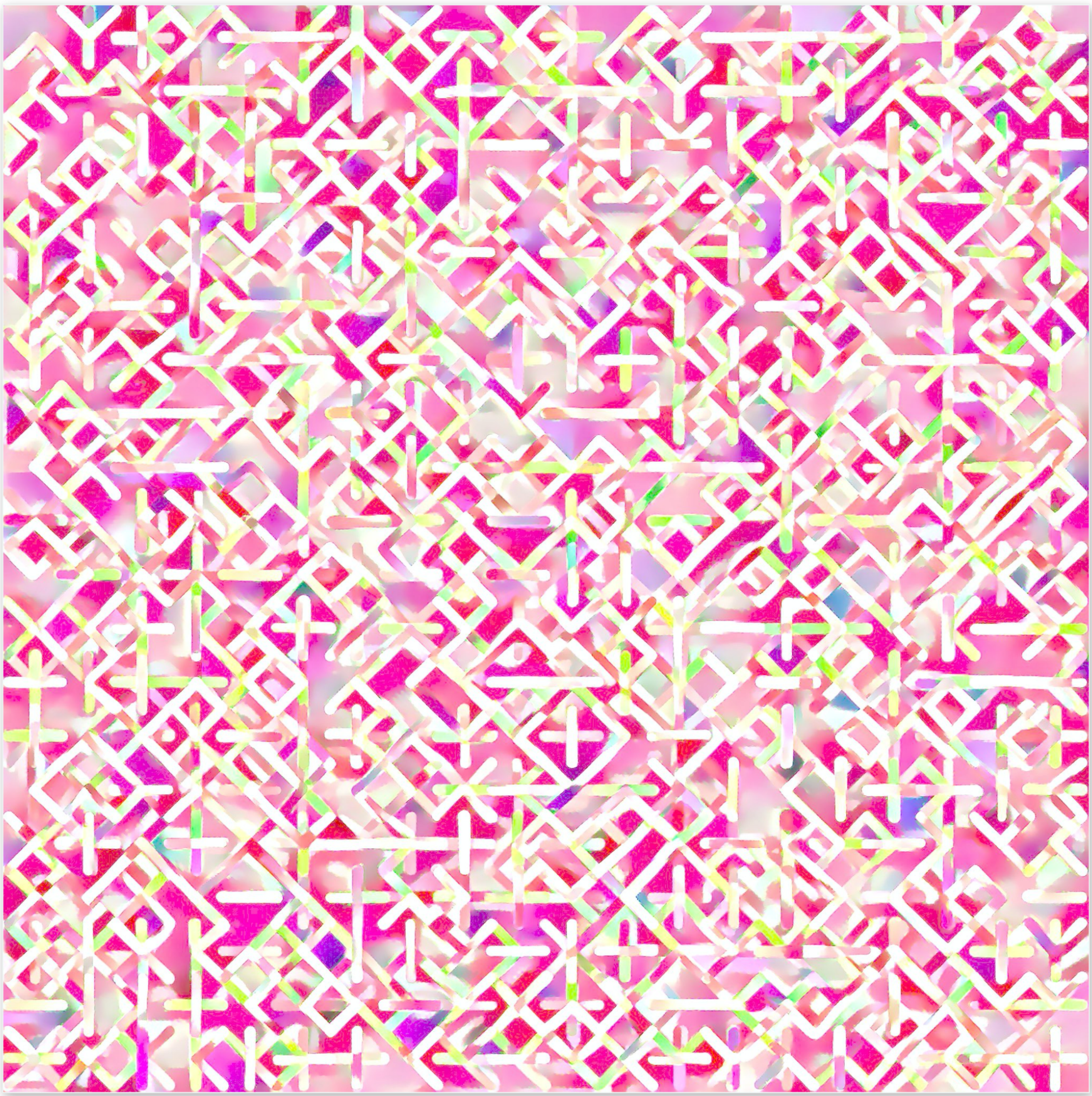
[Figure 13]
Description: Line Grid - Ambiguous Boundary (2019).

A Picture Drawn with One Line of Code

The production process of 'Code Painting' can be expressed with a single line of code. Below is the corresponding one line of code:

```
imgL = Table[ f [imagination, Param->p, Author->"Joo-Haeng Lee"], {p, paramL}]
```

To those seeing computer code for the first time, it may look like a strange foreign language, but its meaning is not complex. It is a process of repeatedly executing function f with Table to output images (imgL). Let's look at it in a little more detail.



[Figure 14]
Description: Line Grid - Spring (2020).

The symbol f is a function. It corresponds to f(x)=y learned in math class. To draw a picture with function f, input is needed, and the most important input is imagination. Creating the function f to implement this imagination so that it can be sensed is one of the most important tasks a 'Code Painting' artist must do. Specifying the author (Author-> "Joo-Haeng Lee") as a symbolic meaning also becomes an input to the function. Another important input is the parameter (Param->p) that controls the operation of the function. For example, in *Line Grid - Evolution of Disorder*, specifying the complexity of the pattern corresponds to the control parameter.

The artist's knowledge and intuition play a large role in deciding which control parameters are needed for a specific function. And finding out what constitutes the most appropriate value for that control parameter requires countless repeated experiments. This repetition is performed for different control parameter values (paramL) inside the Table syntax. In this process, errors are discovered, and sometimes mastered to become new expression techniques. As a result of repeated experiments on different arguments, multiple images (imgL) are obtained. Selecting a work from the images obtained this way is also the artist's role. You can choose a work for an exhibition, or you can choose a commercial work. And that choice will be different for each artist.

If you cannot create the function yourself, you can also utilize a function created by AI. This process is generally the same as the process of experimentation that artists must go through to master new tools and techniques. Furthermore, it is a creative process of implementing new artistic imaginations and concepts so they can be sensed as artworks. This aligns with Picasso's definition of his work as research:

> "Paintings are but research and experiment. I never do a painting as a work of art. All of them are researches. I search constantly and there is a logical sequence in all this research."
> — Pablo Picasso

'Code Painting' can be felt as a cold process where the human role is not needed. However, as examined above, when the working process of 'Code Painting' is expressed in code, we realize that the human role is essential, just as in other arts. The role of the human artist is essential in the various processes of imagining, writing functions, selecting arguments, discovering errors, mastering new techniques, and selecting works from images.

Now you will understand that 'a picture drawn with code' is completely different from 'a picture drawn by code.' Code is just a tool used by the artist, like paint, brushes, and canvas. Therefore, 'a picture drawn with code' is called 'Code Painting' in English. It is in the same context as 'watercolor painting.' For me, traditional computer graphics code was a long-time tool. Now, I am welcoming functions created by AI as a new tool.

Recently, there is a controversy about whether AI can become an artist. I think that if AI can replace all the human roles required in 'Code Painting,' then we might be able to acknowledge AI as an artist. However, I think this is a story of a fairly distant future. Before that, AI will develop into a smart tool for artists. And more and more artists will come to accept AI as a new tool.

As a computer scientist clarifying and developing the principles of AI, and simultaneously as an artist utilizing AI as a tool, I plan to explore the ambiguous boundary between science and art and enjoy myself within it.