

코드로 그린 그림 — 컴퓨터 과학자의 인공지능과 예술 이야기

이주행

(주) 페블러스 공동창업자/대표, ETRI 책임연구원, UST 교수, 작가

들어가는 글

대전 비엔날레에 보낼 작품을 준비하던 중에 인공지능과 예술에 대한 내용으로 기고문 의뢰를 받았다. 연구소에서 인공지능을 연구하는 컴퓨터 과학자이자, 이를 활용해서 틈틈이 작품을 만드는 예술가 입장에서 인공지능과 예술에 대한 나의 경험과 생각을 여러분들께 알릴 수 있는 좋은 기회라고 생각되었다.

글의 앞부분에서는 컴퓨터 그림에 입문하게 된 과정, 그리고 작년에 작가로 데뷔하며 한편으로는 신났고 더불어 좌충우돌했던 개인적 이야기들로 가볍게 시작할까 한다. 후반부에는 비엔날레를 함께 하는 여러분들께 과학과 예술의 만남, 특히 최근 인공지능 기술의 발전이 예술에 주는 의미를 개인적인 경험을 바탕으로 설명드릴 예정이다. 대덕연구단지의 현직 연구원으로써, 또한 대전의 사이언스 페스티벌을 통해 소중한 데뷔의 기회를 얻은 작가의 입장에서, 대전 비엔날레¹가 제시하는 예술의 새로운 방향 설정을 지지하고 응원하는 마음으로 이 글을 준비했다.

—

컴퓨터 그림에 입문하다

나는 대학과 대학원에서 컴퓨터 과학²을 공부하였다. 대학원에서의 세부 전공은 컴퓨터 그래픽스이다. 예술대학에서 배우는 컴퓨터 그래픽스와 달리 공대에서 배우는 컴퓨터 그래픽스는 수학과 물리학에 기반을 둔다. 모양과 움직임과 색을 표현하는 수학적 물리학적 모델을 만들고 이를 컴퓨터에서 시뮬레이션하게 된다.

컴퓨터 그래픽스와의 인연은 사실 컴퓨터 음악에서 시작하였다. 대학 기숙사에서 작은 컴퓨터 음악 실험을 하다가 대학원 컴퓨터 그래픽스 연구실의 애니메이션 프로젝트에서 배경음악을 만드는 일에 참여하게 되었다. 훌륭한 연구장비와 선배님들의 훈훈함도 좋았지만, 컴퓨터로 애니메이션을 만드는 작업에 참여한다는 사실에 매료되었다. 결국 학부 연구참여를 거쳐 컴퓨터 그래픽스 연구실로 대학원 진학을 결심하게 되었다.

¹ 대전비엔날레 2020 《인공지능 : 햇살은 유리창을 잊고》 (Daejeon Biennale 2020 : A.I., Sunshine Misses Windows), 2020. 9. 8. ~ 12. 6. 대전시립미술관, KAIST 비전관.

² 컴퓨터 과학을 영어로는 Computer Science 라고 하는데, 이를 국내에서는 ‘전자계산학’이라고 보통 부른다.

하지만, 막상 연구실 소속이 되고 본격적으로 연구에 참여하면서부터는 애니메이션 작업의 비중은 줄어 들었다. 대신 본격적으로 ‘그림’을 연구할 수 있게 되었다. 중고등학생 시절 기회가 없어서 그림을 제대로 배운 적은 없었지만, 개인적으로 그림 그리기를 즐기고 미술사를 꽤나 즐겁게 공부한 편이었다. 그래서 첨단의 컴퓨터로 그림을 그리는 공부를 할 수 있다는 것은 대단히 설레는 일이었다.

하지만, 본격적인 컴퓨터 그래픽스 연구는 ‘예술’과는 다소 거리가 있다. 심미적으로 연구 결과물을 활용하는 것도 고려하지만, 중요한 것은 깊은 수학적인 이론과 어려운 컴퓨터 알고리즘들을 섭렵하는 것이다. 지도 교수님께서는 수학을 전공하시고 컴퓨터 과학을 복수로 공부하신 분이라 컴퓨터 그래픽스에 대한 이론적 깊이가 대단하셨다. 교수님과의 공부 덕분에 보통의 컴퓨터 과학 전공 대학원생들과 달리 꽤나 고급스러운 수학의 세계를 경험할 수 있었다. 대학원을 졸업한지 꽤나 오랜 시간이 흘렀지만 그 때 공부했던 수학과 추상적인 사고의 경험들이 그 이후 공학적 연구와 작품 활동에 큰 도움이 되었다.

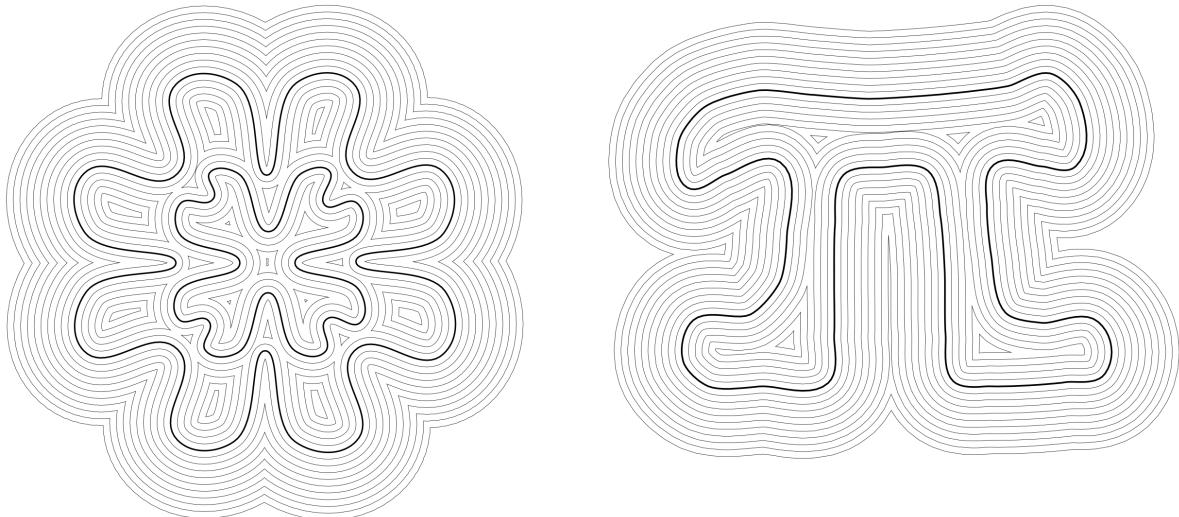


그림 1. 자유곡선의 오프셋 곡선에 대한 박사과정 연구. 곡선을 따라 움직인 서로 다른 크기의 원의 패적을 표현하고 있다. 산업 현장에서 매우 중요한 의미를 갖지만, 그 형태만으로도 매력이 있다. (1999, POSTECH)

아무리 이론으로 무장한다고 해도 컴퓨터 그래픽스 연구의 결과는 역시 아름다운 수식과 더불어 멋진 그림이 더해져야 한다. 기존의 이론으로 아름다운 그림을 만들어 보고, 더 나은 그림을 만들 수 있는 새로운 이론을 찾아 도전하는 것은 대단히 매력적인 과정이었다. 또한 당시 최고의 컴퓨터 그래픽스 장비를 이용해서 멋진 그림을 마음대로 그려볼 수 있다는 것은, 어려운 이론 공부에 대한 달콤한 보상 이기도 했다. 정말이지 컴퓨터 그래픽스 전공은 나에게 있어 연구와 취미가 하나가 된 듯한 매우 이상적인 분야였다.

지금 생각해 보면 내가 학교에서 주로 배운 것은 ‘현실을 관찰하여 모델링하고 컴퓨터로 모사³(模寫)하는 방법’이었다. 모사의 결과가 좋을 때는 아름다운 영상으로 표현되니까 시각적 호사를 덤으로 누릴 수 있었다. 하지만, 결국 중요한 것은 복잡한 현실을 모델링하기 위해 필요한 추상적인 사고의 과정과 수학과 물리학의 도구들이었다. 이러한 모델링과 시뮬레이션에 대한 경험과 수학과 물리학에 대한 지식은, 대학원에서 학위를 마치고 연구소에서 일하면서도 대학원 전공 분야 외의 다양한 분야를 두루⁴ 연구해 볼 수 있는 토대가 되었던 것 같다.

-

코드로 상상을 그리다 – 무엇을 어떻게 그릴까?

대학원을 졸업하자마자 병역특례로 연구소에 입사하게 되었다. 연구원 생활 초기에는 바쁘고도 즐거운 나날들을 보냈다. 무엇보다 대학원에서 공부한 지식을 기반으로 새로운 실험들을 다양하게 해 볼 수 있었다. 특히 당시 내가 근무하던 연구팀은 정부출연연구소의 보통의 분위기와는 달리 매우 원활한 소통과 자유로운 연구가 ‘묵시적’으로 보장되었다. 그 덕분에 월급을 받는 대가로 수행하던 프로젝트들 외에 좀 더 원천적인 연구주제들을 다루며 논문을 작성할 수 있었다. 덕분에 ‘가현학술상’과 같은 굽직한 연구상을 두 번이나 수상할 수 있었다고 생각한다.

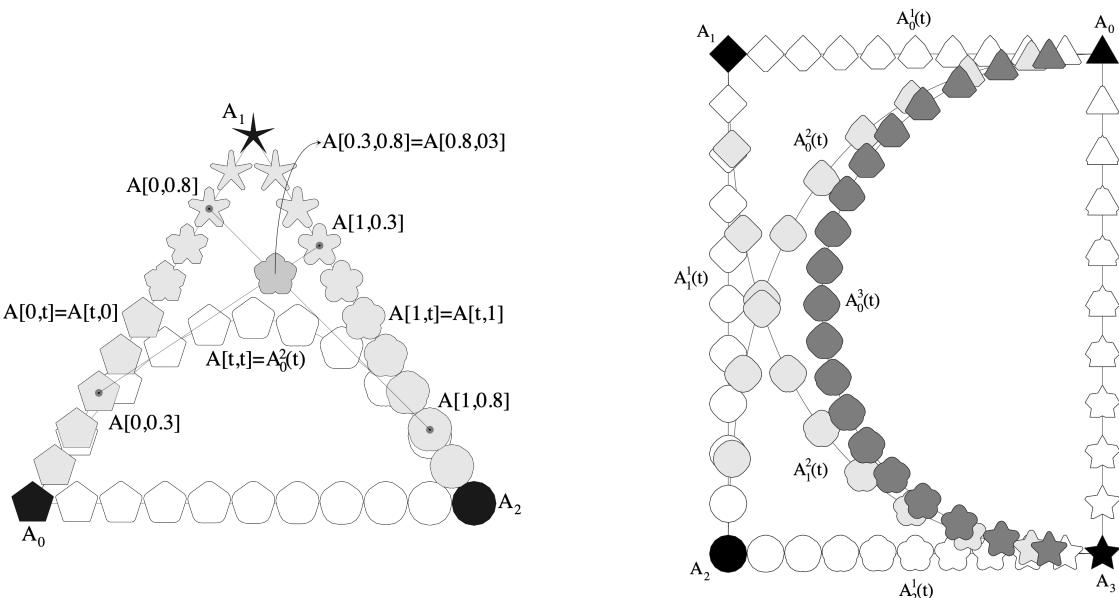


그림 2. 다각형의 방향지도(Direction Map)를 이용한 형태 혼합(Shape Blending)에 관한 연구 (2003)

2000년대 초반에서 2010년도 중반까지의 내 연구의 기반은 수학과 물리학적 지식을 기반으로 한 모델링과 시뮬레이션이었다. 로봇에서 렌더링 기술까지, 연구 주제는 달라도 방법론은

³ 모사는 시뮬레이션(simulation)이라고도 한다. 실제로 로봇을 만들지 않아도 그 움직임을 미리 관찰할 수 있고, 실제로 페인트를 만들지 않아도 그 광택의 효과를 그려 볼 수 있는 것이 모사의 예이다.

⁴ 내 이름 ‘주행(周行)’에 있는 ‘주’(周)자 덕분에, 다양한 분야를 두루두루 탐색하고 즐기고 있다고 생각한다.

동일했다. 그리고 연구결과⁵ 발표를 위해서는 언제나 매우 세심하게 준비된 그림들을 사용하였다. 더불어 발표자료 작성에도 매우 많은 공을 들였다. 이는 기본적으로 대학원에서 무수히 훈련 받은 바이기도 하지만, 내용과 형식의 모두에서 완성도를 추구하는 개인적인 성향이기도 했다. 언제부터인지 모르겠지만, 논문 발표는 청중을 위해 잘 준비된 ‘짧은 연극’ 또는 ‘쇼’라는 생각을 하게 되었다. 물론 논문의 그림, 발표 자료, 발표 그 자체가 모두 객관적인 지식 소통의 수단이지만, 아무래도 나의 심미적인 취향이 반영되었을 것으로 생각된다. 이러한 배경은 오랜 습작 활동과 최근의 작품 활동에 큰 도움이 된 것 같다.

컴퓨터로 그림을 그리는 도구는 매우 다양하다. 경우에 따라 선택하는 도구들이 다르다. 포토샵과 같은 이미지 도구를 최종단계에서 쓰기도 하지만, 컴퓨터 그래픽스를 전공한 공학자의 그림은 대부분 ‘코드’로 만들어 낸다. 코드(code)란 ‘컴퓨터 프로그램’이라고도 한다. 사람들이 소통을 위해서 언어를 사용하는 것처럼, 우리가 컴퓨터에게 일을 시킬 때는 특별한 언어가 필요하다. 이를 프로그래밍 언어라고 한다. 코드는 프로그래밍 언어로 작성한 특별한 ‘글’이다. 가끔 아름다운 코드도 있지만, 대개는 매우 길고 난해하여 지루한 설명글이다. 하지만 이런 코드를 통해 사람의 생각을 컴퓨터에게 전달할 수 있다. 인공지능 기술의 발전으로 요즘의 컴퓨터는 간단한 입력에도 알아서 척척 답을 주지만, 아직도 대부분의 경우에는 매우 꼼꼼하게 지시를 해야만 원하는 답을 얻게 된다.

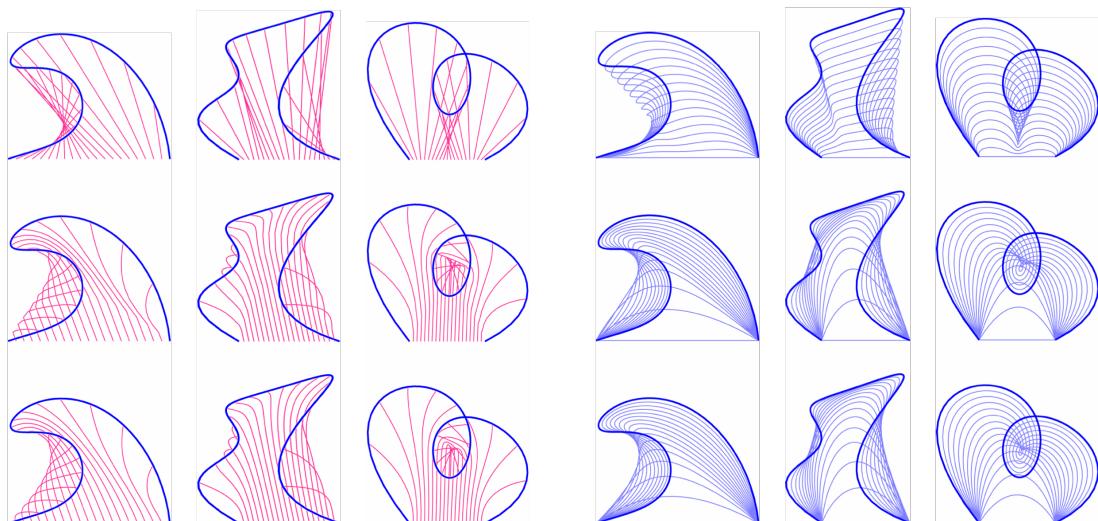


그림 3. 고차 베지어 곡선의 성장 구조 (Rib and Fan). 서로 다른 곡선마다 유일한 ‘지문’이 있을까라는 상상에서 출발한 기하 연구 (2006)

이렇게 사람이 입력한 코드에 따라 컴퓨터는 작업을 수행하고 그 결과를 출력한다. 문자, 소리, 움직임의 출력도 있겠지만, 컴퓨터 그래픽스에서는 주로 컴퓨터가 만들어내는 이미지 출력을 다룬다. 출력된 이미지를 보면 코드의 오류가 보이기도 한다. 공학적인 연구에서 그림을 그리는

⁵ 지면 관계상 자세한 연구 내용은 생략한다.

경우는 대개 미리 정해진 목적이 있다. 만약 그림에 이 목적에 어긋나는 요소⁶가 있다면 오류가 있는 것으로 간주하고, 그림을 만드는 코드를 수정하게 된다.

코드를 작성하고 그림을 확인하는 과정은 여러 번 반복하는 것이 보통이다. 이 과정은 대개 매우 긴 시간이 걸리고 꽤 지루하기도 하다. 따라서 코딩은 일종의 지식노동 과정이라고 볼 수 있다. 그럼에도 불구하고 좋은 결과를 얻을 수만 있다면, 노동에 대한 충분한 보상이 될 수 있다⁷. 하지만, 과정 자체가 즐거운 코딩은 불가능할까? 경감된 노동으로 아름다운 그림을 만들 수는 없을까? 마치가 화가가 물감을 묻힌 붓을 캔버스에서 움직이며 작업하는 것처럼 즉시적이고 직관적일 수는 없을까? 이는 어떤 프로그램 언어를 사용하는가에 따라 큰 영향을 받는다. 실제로 작동하는 큰 시스템을 만들 때는 C나 Python 같이 엄밀한 프로그램 언어를 사용하게 된다. 하지만, 창의적인 그림을 그릴 때 이런 프로그램 언어를 사용한다면 그림을 그리고 싶은 감흥이 깨질 수도 있다. 준비도 복잡하고 인터랙티브한 피드백도 약하기 때문이다.

반대로 울프램 언어(Wolfram Language)같은 프로그램 언어도 있다. 울프램 언어는 매스매티카(Mathematica)라는 특별한 소프트웨어 위에서 동작한다. 이 언어는 과학자들이 문제 해결에 집중할 수 있도록 만들어 진 컴퓨터 언어로서, 직관적인 코딩이 가능한 것이 큰 장점이다. 특히, 프로그램을 위한 복잡한 준비 없이 머리 속에 떠오른 아이디어를 빠르게 스케치해서 실험해 볼 수 있다. 마치 커피를 먹다가 냅킨에 아이디어를 그리는 것과 같다. 컴퓨터 기반 과학의 기본 방법인 모델링과 시뮬레이션을 위한 기능도 뛰어나지만, 그림을 그리기 위해서만 사용해도 매우 유용하다. 심지어 철학자, 작가들이 자신의 아이디어를 정리하고 전달하는데 사용하기도 한다.

⁶ 달 표면의 특징을 가시화하려고 있는데 거친 난반사의 느낌 대신 유리처럼 반짝임이 보인다면 재질 표현에 오류가 있는 것이다. 물론, 동화책에 사용할만한 충분히 재미있는 그림임에도 불구하고 말이다.

⁷ 하지만, 인공지능이 코드를 만드는 것을 경험하고 나면 생각이 바뀌기도 한다.

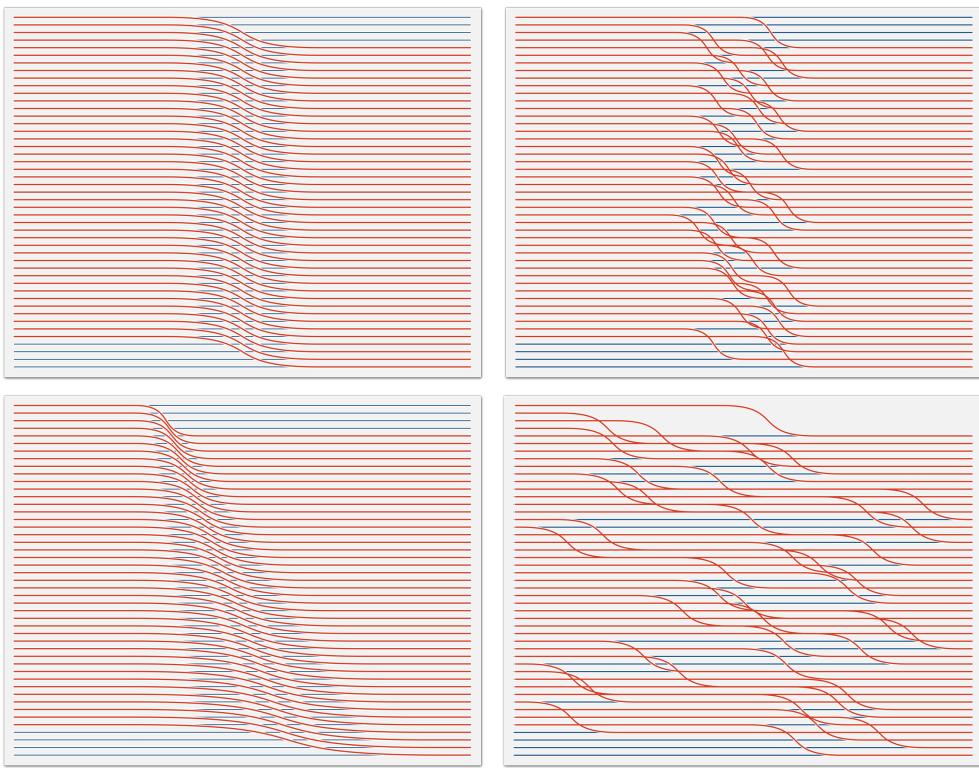


그림 4. 마음의 변화를 기하학적 기호로 표현한 예. 연결된 선 – 네 가지 흐름 (Connected Lines – 4 Streams, 2017)

나는 대학원 때부터 매스매티카를 사용해 왔다. 대부분 지식 전달의 목적을 갖는 그림을 그리는 용도였다. 하지만, 도구의 편리성 때문인지 점점 ‘엉뚱한’ 작업에도 사용하기 시작했다. 바로 ‘코드로 그린 그림’의 시작이다. 지식 전달을 위해서 그렸던 그림들 외에 상상과 마음을 담은 그림들도 그리게 되었다. 두 종류의 그림들은 서로 내용이 달랐지만 그 제작 방법과 형식은 동일했다. 공부를 하다 보면 공책 한 구석에 낙서를 하기도 하지 않는가? 딱 그러한 과정이었다. 더 멋있게는 문인화에 비유해 주신 분들도 계셨다. 옛 선비들이 글공부를 하던 붓과 먹과 종이로 관념의 세계를 그리거나 유희를 기록하지 않았던가. 연구를 하던 코딩으로 그림을 그렸으니 확실히 맥락이 달아 있다.

—

인공지능⁸을 만나다 – 새로운 불의 발견

수학과 물리학의 이론, 그리고 컴퓨터 모델과 시뮬레이션 방법을 활용한 시각적 표현은 2015년 무렵까지도 나에게 가장 중요한 연구 주제였다. 어떠한 현상이든지 간결한 수식과 기호로 표현하는 것을 추구하던 시기였다. 예를 들어, 카메라의 원리에 대한 기하학적 접근을 한 경우가 있었다. 이미 오래된 연구 분야이지만, 새로운 질문 하나를 찾게 되었고, 그 답을 찾기 위해 여러가지 시도를 했다. 좀 더 자세히는, 종횡비(aspect ratio)가 알려지지 않은 직사각형 한 장이

⁸ 인공지능(AI, Artificial Intelligence)이 다루는 범위는 역사적으로나 주제적으로 매우 광범위하다. 이보다는 기계학습(Machine Learning), 더 세부적으로는 딥러닝(Deep Learning)이 기술적으로는 더 정확한 표현이다. 하지만, 편의상 인공지능으로 적는다.

포함된 카메라 이미지에서 그 직사각형의 정확한 모양과 카메라의 위치와 회전을 동시에 알아내는 문제였다. 머리 속의 막연한 질문을 수학적 형식에 맞게 기술하고, 그 해답을 찾기 위해 수 많은 가시화와 수식 정제의 과정을 거쳤다. 마침내 찾아 낸 수식은 간결했고 아름다웠다.

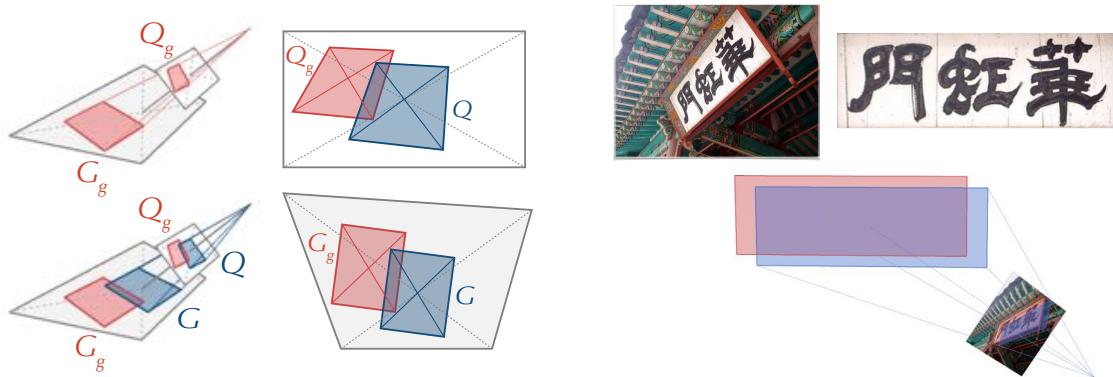


그림 5. 직사각형과 카메라의 관계에 대한 기하학적 연구. 형태를 정확히 모르는 직사각형에 대한 한 장의 사진으로 그 직사각형의 정확한 모양과 카메라의 촬영 위치를 한번에 알 수 있다. (2012, 2013)

하지만, 최근에는 이러한 틀에서 벗어나는 새로운 관심분야가 생기게 되었다. 난생 처음 보는 별종이 나타났는데, 도저히 이 녀석에게 눈길을 뺄 수가 없었다. 그래서 자꾸 바라보다가 그 치명적인 매력에 빠지게 되었다. 그런데, 완벽한 줄 알았던 이 녀석을 가까이서 관찰해 보니 허점도 가득했다. 많은 사람들이 ‘4 차산업혁명’의 중심에 이것이 있다고 하고, 이것 때문에 수많은 일자리가 위협받게 될 것이라고도 한다. 바로 인공지능(AI, artificial intelligence)이다.

그렇다면 나는 인공지능의 어떤 매력이 나의 관심을 끌었을까? 기존에 좋아하던 수학적 방법과는 크게 다른데도 불구하고 말이다. 아마도 인공지능 덕분에 새로운 차원의 코딩 방법을 경험했기 때문인 것 같다. 고도의 지식을 갖추고 충분한 훈련을 받은 인간이 한정된 범위의 문제에 대한 해답을 기호들의 정교한 상호작용으로 찾아내는 것이 기존의 차원이었다면, 고성능 컴퓨터가 충분한 데이터에서 찾은 비정형적 패턴들의 조합을 통해 누구라도 거대한 문제의 해답을 찾을 수 있는 것이 새로운 차원이었다. 비록 이 새로운 차원의 접근법 어떻게 동작하는지 아직은 정확히 알 수 없고, 그 동작도 완벽하지는 않지만 말이다.

복잡한 현상을 관찰하여 이를 설명하는 간결한 수식을 찾아내고, 이를 손수 코드로 구현하여 아름다운 그림을 만드는 것은 우아하고도 지적인 과정이다. 코드를 만드는 데는 비록 수작업이 필요하지만, 이 역시 코딩 장인의 손길이 한 땀 한 땀 필요한 분야라는 자부심도 있다. 이런 작업 방식을 사랑하던 사람에게, 최근의 인공지능 방법은 꽤나 놀랍고 낯설고 또한 당황스러웠다. 서울에서 전학 왔다는 잘 생긴 부잣집 친구가 첫 시험에서 1등을 해버렸는데 말할 때는 맹구 느낌이 나는 상황이랄까? 이 느낌을 공유하고 싶으니, 좀 더 설명해 보겠다.

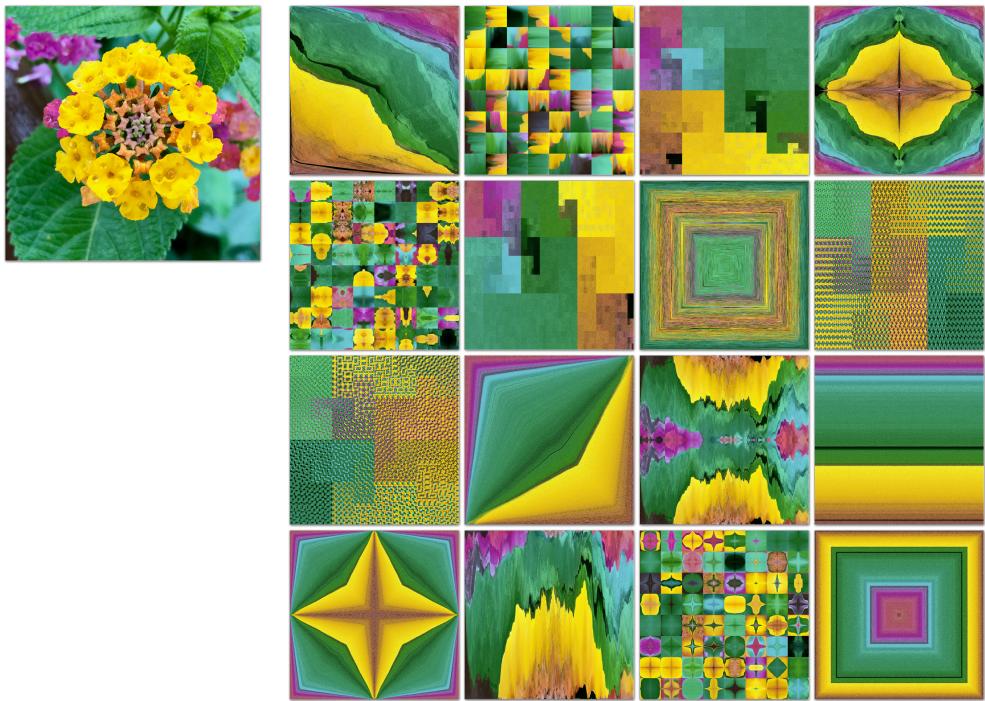


그림 6. 기호의 재구성으로 만들어 낸 패턴의 예. '란타나'와 4x4 픽셀스택 (Lantana and 4x4 Pixel Stack, 2018)

'코드로 그린 그림'에서 인간의 역할을 생각해 보자. 코드는 컴퓨터가 실행하지만, 그 코드를 만드는 것은 사람의 역할이다. 우선 컴퓨터가 할 일을 정의해야 하다. 먼저 입력과 출력에 대한 정의가 필요하다. 그 다음은 주어진 입력에서 출력을 만들기 위한 작업 절차가 정의되어야 한다⁹. 컴퓨터는 그 절차를 전혀 모른다. 대신 사람이 이를 정확히 알고 있어야 한다. 이렇게 입력, 출력, 절차가 정의되면 이를 그대로 코드로 옮기는 과정이 필요하다. 똑똑한 사람에게 일을 시킬 때와 달리 컴퓨터에게 일을 시킬 때는 지나치다 싶을 정도로 '친절하게' 설명을 해 줘야 한다. 코드에 적어 준 그대로만 동작하기 때문이다. 그래서 코딩은 높은 숙련도와 많은 시간과 노력이 필요한 분야이다.

⁹ 컴퓨터 과학에서는 이러한 작업 절차를 알고리즘(algorithm)이라고 한다.

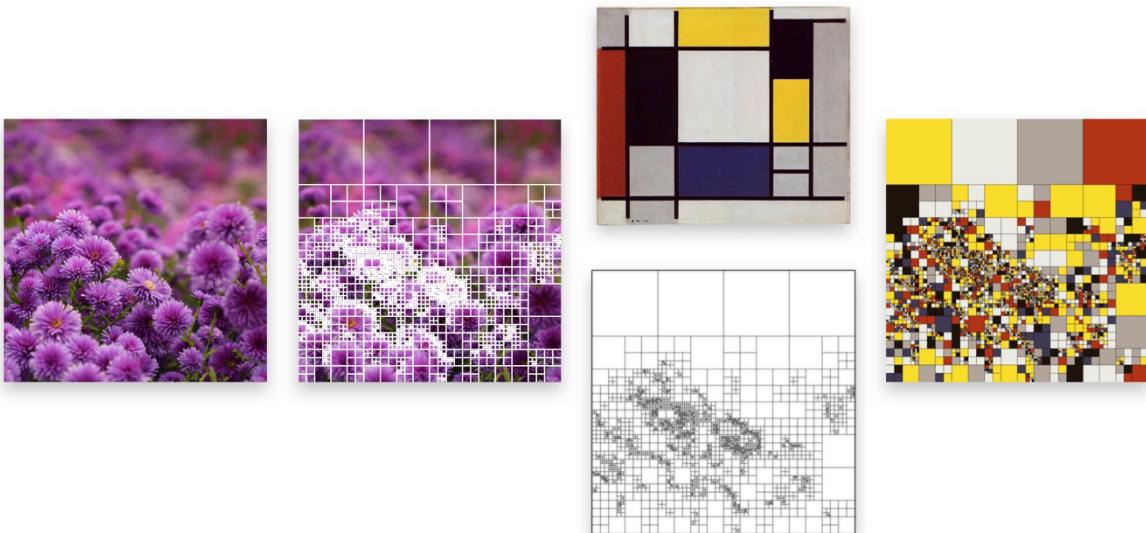


그림 7. 사진에서 몬드리안 스타일의 구성을 자동 생성하는 실험. (추상의 탄생, 2018)

이 맥락에서 인공지능의 놀라운 점은, 작업의 입력과 출력 정의만 되어있고 그 절차가 정의되지 않아도 그 일을 스스로 해낸다¹⁰는 점이다. 이걸 ‘지능’이라고 부르지 않는다면 다른 무엇이 지능이겠는가? 예를 들어, 백만 장의 서로 다른 사진을 천 개의 범주(class)¹¹로 분류하는 일은 입출력에 대한 정의가 충분히 가능한 일이다. 하지만, 실제로 범주 분류(classify)에 필요한 자세한 절차를 명시하는 것은 인간에게는 거의 불가능하다. 따라서, 이에 해당하는 코드를 작성하는 것 역시 불가능하다¹². 하지만, 인공지능은 작업 정의가 주어지고 이에 부합하는 데이터만 있으면, 작업 정의에 따라 수행되는 코드를 스스로 만들어 낼 수 있다. 인간이 자세하게 절차를 설명해 주지도 않았는데 말이다¹³! 다만 필요한 데이터의 양이 어마어마하게 많아야 하고, 코드를 만들기 위한 특별한 기계학습(machine learning) 방식과 계산 장치가 필요하다.

¹⁰ 정확히는 그 일을 해내는 코드를 스스로 만들어 낸다.

¹¹ 이미지에 포함된 고양이, 나비, 자동차 등이 범주에 해당한다. 물론 여러 개의 범주들이 한 이미지에 함께 있기도 하다.

¹² 범주화 작업을 위한 가장 중요한 정보는 특징(feature)인데, 예를 들어 서로 다른 고양이들이 공통적으로 갖고 있는 외형 특징을 어떻게 수학적으로 정의하고 이를 코드로 표현할 수 있단 말인가?

¹³ 절차를 명시하지 않는다고 해도, 현재의 인공지능에서도 인간의 개입은 상당히 필요하다. 그 개입에 일반적이고 체계적 가이드가 존재하지 않고, 여러 번의 시도에 의존해야 한다는 것이 문제이다.

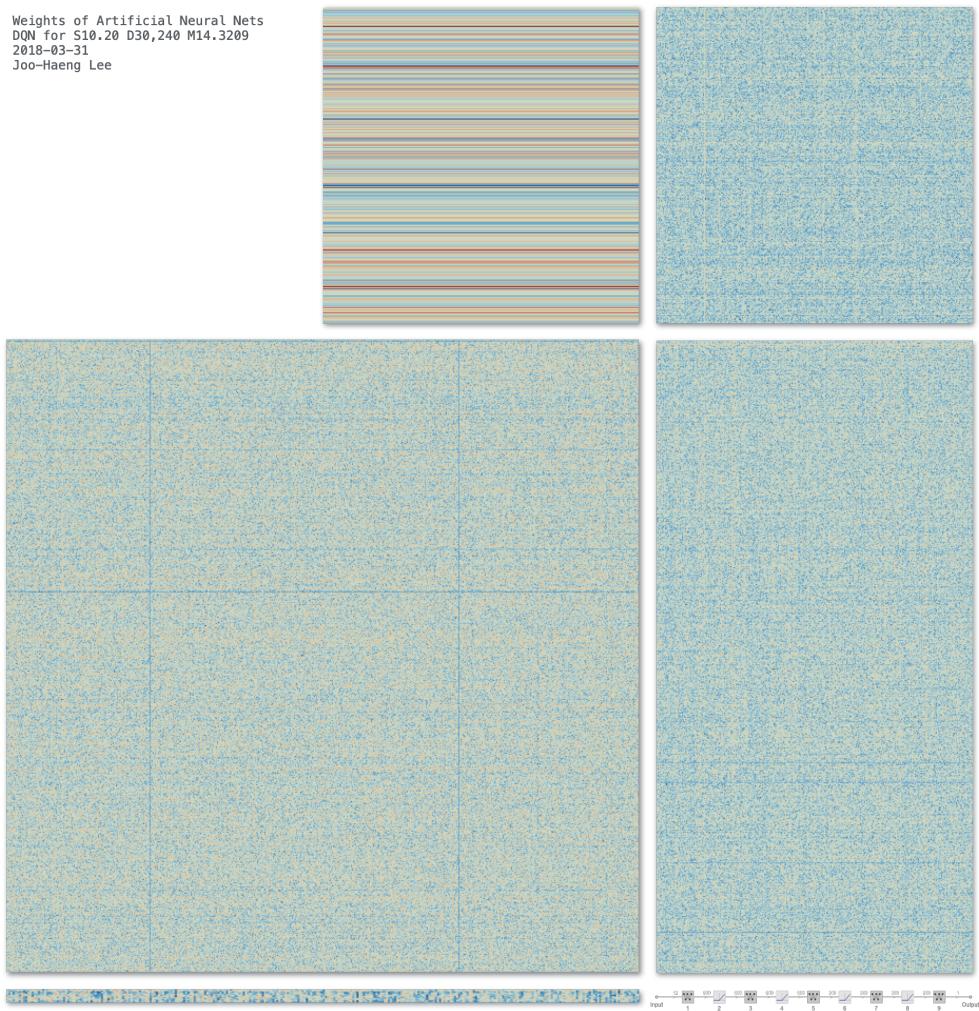


그림 8. 심층강화학습 신경망의 가시화의 예. 스트림스 게임 S10.20 의 경우. (2018)

이렇게 주어진 데이터에서 인간의 개입없이 컴퓨터가 스스로 코드를 만들어 내는 방식은 인공지능의 오랜 주제였다. 다양한 방법 중에서 최근의 시도들은 인공신경망(ANN, artificial neural network)과 직접적인 관계가 있다. 내가 인공신경망을 처음 접한 것은 1990년대 대학원 시절이었는데, 이론적으로 그다지 우아함을 느끼지 못했고, 실제로 동작하는 것도 당시에는 매우 작은 문제에 국한되어 있었다. 그래서 인공신경망은 나에게는 전혀 관심 밖의 연구 주제였다.

나뿐만 아니라, 당시의 많은 연구자들이 인공신경망을 외면하였다. 하지만 소수의 해외 연구자들은 꾸준히 인공신경망 기법을 발전시켰고, 오늘날 심층신경망(DNN, deep neural network)이라는 기법으로 발전하게 되었다. 개념적인 면에서 심층신경망은 기존의 인공신경망과 크게 다르지 않지만, 신경망의 복잡도와 규모면에서 엄청난 증가가 있었다. 따라서 다룰 수 있는 데이터의 양과 종류도 크게 증가하였다. 또한 신경망을 구동하는데 필요한 GPU¹⁴와 같은 특별한 계산장치의 발전도 함께 하였다. 즉, 알고리즘, 데이터, 하드웨어, 이렇게 3 박자가 잘 맞은 상황은

¹⁴ Graphic Processing Unit. 원래는 컴퓨터 그래픽스의 계산 가속을 위해 고안된 장치인데, 이제는 인공지능 실험을 위한 필수적인 도구가 되었다. 컴퓨터 그래픽스 전공자로서 GPU의 쓰임이 이렇게 변화하는 것을 보면 매우 특별한 느낌이 든다.

심층신경망의 발전에 매우 유리한 상황이었고, 덕분에 기존의 방법으로는 엄두도 못 내던 거대한 문제들을 해결할 수 있게 되었다. 드디어 컴퓨터가 스스로 만든 코드의 시대가 온 것이다.

바둑은 이렇게 인공지능이 해결한 거대한 문제 중의 하나였다. 그래서 알파고의 승리는 심층신경망 기반 인공지능 기법의 존재감을 전세계에 알린 충격적이고도 상징적인 사건이 되었다. 덕분에 인공지능에 매우 냉담하던 컴퓨터 그래픽스 분야를 비롯한, 다양한 분야의 컴퓨터 과학자들이 심층신경망이라는 데이터 기반 코딩법을 진지하게 받아들이기 시작했다. 물리학, 신경과학, 의학, 경제학 등 여러 학문 분야에서도 인공지능이라는 새로운 도구의 활용을 진지하게 고민하고 적극적으로 수용하게 되었다. 나도 이 무렵에 인공지능에 대한 색안경을 벗을 수 있었다. 다만, 기존의 우아한 수학적 방법을 버린다기 보다는 데이터 기반 자동 코딩 방법을 새로운 도구의 하나로 받아 들이게 된 것이다.

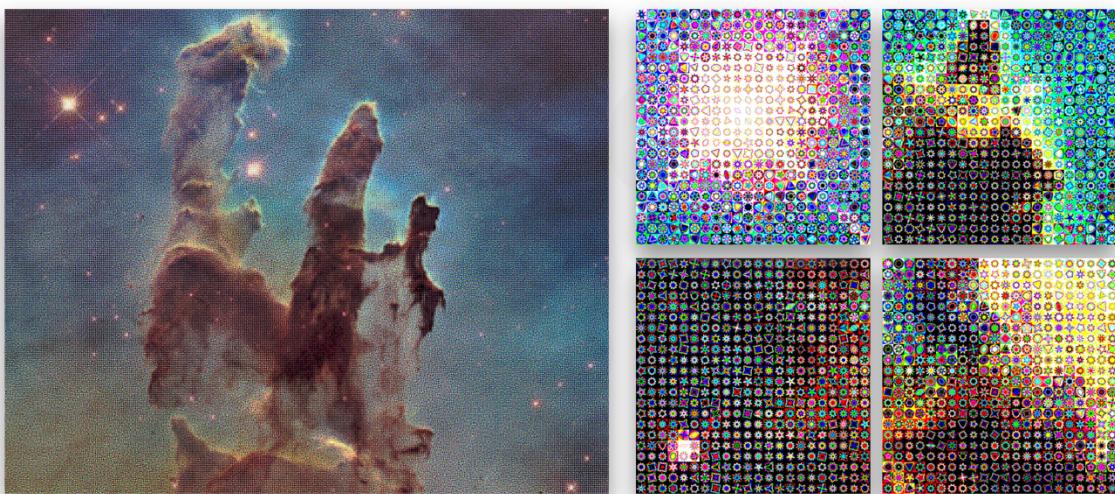


그림 9. 인공지능 실험을 위해 만든 별모양의 가상 학습셋 Star-MNIST 를 이용한 그림 그리기. (Star Swap 창조의 기동 성운, 2019)

하지만, 현재의 인공지능에 대해서는 여전히 조심스러운 입장이다. 학습에 필요한 데이터의 양이 지나치게 방대해야 한다는 점, 탄소 배출과 관련되는 에너지 소모가 지나치게 크다는 점, 그 내부의 작동 원리를 인간이 이해할 수 없다는 점 등 풀어야 할 숙제는 넘친다. 그럼에도 현재의 인공지능의 성취 또한 명확하다. 인류 문명사에서 과거에는 꿈도 꾸지 못했던 거대한 규모의 데이터와 복잡한 문제를 다룰 수 있게 된 것이다. 이것은 마치 인류가 불을 처음 발견했을 때와 유사하지 않을까 생각한다. 불에 구운 고기 맛이 기가 막히다는 것을 처음 알게 되었고, 너나 할 것 없이 불의 화려한 매력에 빠져서 다양한 실험들을 한다. 하지만, 아직 불을 정복하지 못했기 때문에 불을 꺼뜨리기도 하고, 불에 데기도 한다. 그러나 인간은 결국 불을 정복하고 몇 차례의 산업혁명을 거쳐 오늘날의 문명을 이루게 되었다. 그렇다면 인간은 인공지능에 대해서도 그 원리를 규명하여 정복하고, 이를 발판삼아 다음의 문명으로 나아갈 수 있을까?

오류를 즐기다 – 넓어진 감각의 지평

이런 질문을 배경으로 하고, 인공지능이 나의 ‘코드로 그린 그림’ 작업에 준 영향을 설명해 볼까 한다. 현재의 인공지능은 기존에 컴퓨터 그래픽스 기술이 제공하지 못하던 많은 기능에 도전하고 있고, 일부에서는 이미 기존 기법을 능가하였다¹⁵. 나는 여러가지 가능성 중에서도, 인공지능을 이용하여 복잡한 패턴을 쉽게 만들 수 있는 것이 가장 마음에 듈다. 하지만, 새로운 도구만을 고집하고 있지는 않다. 기존에 수학적 방법으로 그리던 그림이 수채화라면, 인공지능 코드로 그린 그림은 유화로 볼 수 있다. 전혀 다른 재료들이지만 회화의 도구라는 점에서는 공통적이고, 필요에 따라 달리 선택하기도 하고 혼용하여 쓸 수도 있기 때문이다.

그리고, 인공지능은 그림을 그리는 색다른 도구를 넘어서, 새로운 표현의 가능성을 열고 창작의 영감을 주는 혁신적인 도구라는 점이 더 중요하다. 예를 들어, 이번에 비엔날레에 전시하는 ‘라인 그리드(Line Grids)’ 시리즈는 수학적 도구를 이용한 기호에서부터 인공지능을 이용한 패턴까지, 다양한 표현법을 담고 있다. 특히, 이 시리즈의 후반부에 해당하는 복잡한 패턴에 기반한 작업은 인공지능 방법이 아니었으면 불가능했을 것이다. 덕분에 새로운 비정형적 패턴 생성 방법을 찾을 수 있었고, 이렇게 만든 작품을 통해 정형적 기호와 비정형적 패턴 사이의 관계를 새롭게 생각해 보는 계기가 되었다.

¹⁵ 애니메이션과 렌더링 분야는 수학적 이론의 깊이가 남다른 분야로, 인공지능이 사용될 것으로 상상하지 못했었다. 요즘은 딥러닝을 통해 캐릭터 애니메이션을 만들수 있을 뿐만 아니라, 렌더링에도 활용되고 있다.

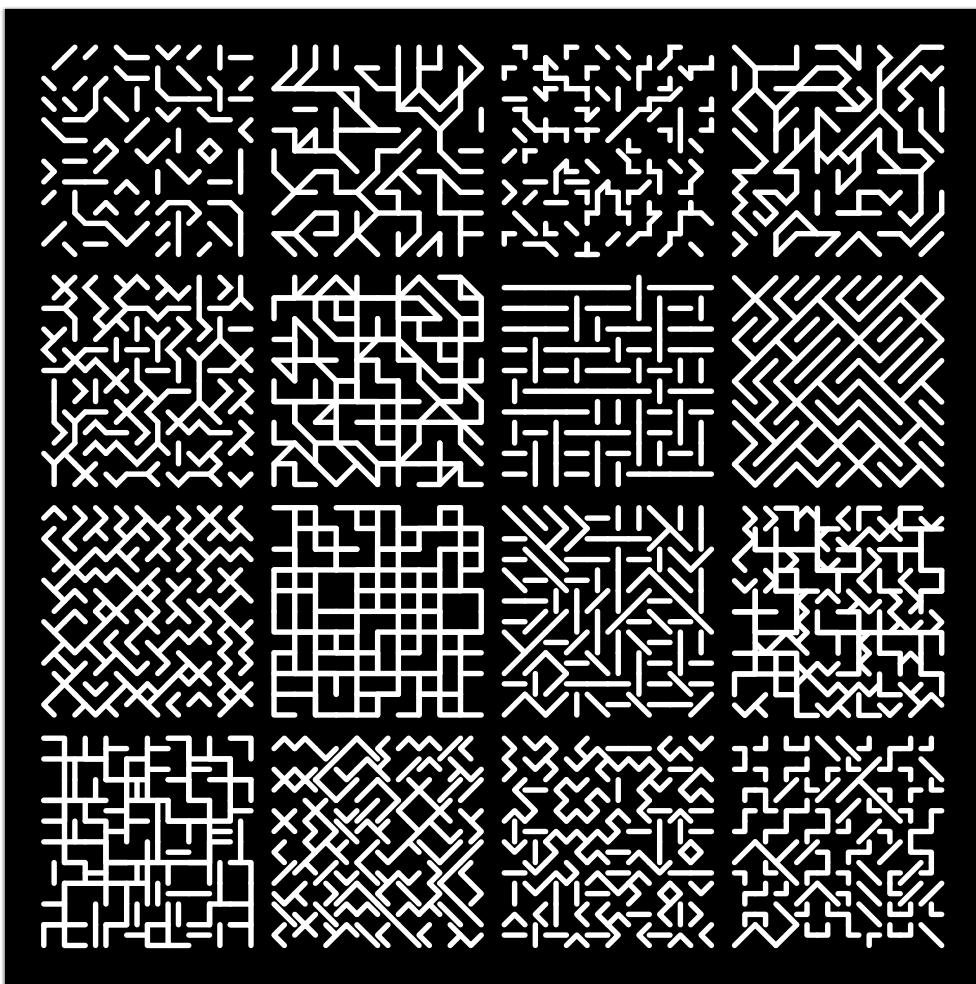


그림 10. 선분의 반복으로 만든 기하학적 패턴의 예. 라인그리드 – 16 종족 Atlas of Line Grids (2018)

‘라인 그리드 16 종족(Atlas of Line Grids – 16 Tribes)’은 추상적인 상상과 개념을 기하학적인 기호로 표현한 그림이다. “단순한 선분이 복잡한 패턴의 유전자가 될 수 있는가”라는 질문에서 출발하여 선분의 반복과 중첩으로 새로운 패턴을 만들어낸 실험의 결과이다. 이런 작업 방식이 인공지능을 만나기 전에 내가 즐겨하던 방식이다. 하지만, ‘라인그리드 – 무질서의 진화(Line Grids – Evolution of Disorder)’의 제작에서부터는 인공지능 기법을 활용하기 시작했다. 기존의 기하학적인 기호를 활용하여 만들어 낸 밑 그림에 인공지능 기법을 추가로 사용하여 복잡한 패턴을 만들고 이를 작품 제작에 활용하였다. 인공지능이 만든 코드에게 작품 제작의 모든 것을 맡기는 방식과는 다르다. 아래에서는 이 방법을 조금 더 살펴보자.

이미지를 다루는 인공지능 기법 중에 ‘스타일전이(Style Transfer)’라는 방법이 있다. 보통 스타일전이에는 두 개의 입력 이미지가 필요하다. 하나는 스타일 이미지로 그림을 그리는 형식을 담고 있다. 다른 하나는 콘텐츠 이미지로 그림의 내용을 담고 있다. 예를 들어, 고흐의 이미지를 스타일로 넣고, 꽃 사진을 콘텐츠로 넣어 주면, 스타일전이 심층신경망은 고흐의 화풍을 모방하여 꽃 사진을 변형하게 된다. 마치 고흐가 그런 듯한 꽃을 만들어 내는 것이다.

나는 여기에 재미있는 시도를 해 봤다. 스타일과 콘텐츠 입력으로 동일한 라인그리드 패턴 이미지를 넣어 준 것이다. 만약 이러한 상황에서 사람이 직접 스타일 전이를 계산해야 한다면, 두 입력 그림이 같다는 것을 쉽게 알아 봤을 것이고, 입력 이미지 중 하나를 그대로 출력했을 것이다. 하지만 인공지능 스타일전이는 스타일과 콘텐츠가 동일함을 인식하지 못했다. 대신 입력으로 받은 정형적인 패턴을 불규칙한 패턴으로 변형시키기 시작했다. 더 나아가 불규칙의 정도도 제어가 가능해 보였다.

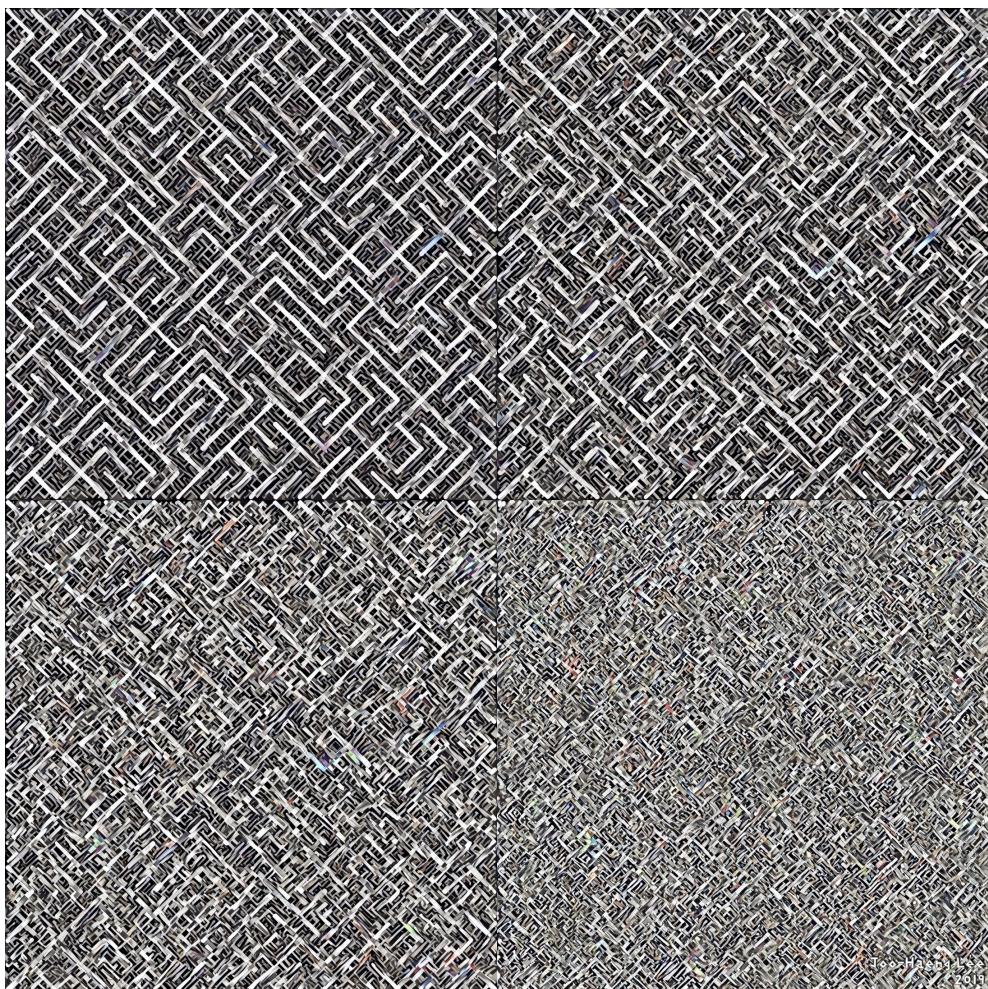


그림 11. 스타일 전이를 이용한 패턴 생성의 예. 기학적으로 정의된 정형적 패턴을 딥러닝을 통해 단계적으로 ‘무너뜨리는 과정’을 담고 있다. 라인그리드 – 무질서의 진화 (Line Grid – Evolution of Disorder, 2018)

이는 흥미로운 발견이었다. 입력과는 유사하지만 뭔가 달라진 패턴이 만들어질것으로 예상했지만, 그 구체적인 모습을 머릿속으로 상상할 수는 없었다. 결과적으로 이 실험을 통해 정형적인 기하학적 기호로부터 비정형적인 패턴을 만드는 매우 간단한 방법을 찾게 되었다. 예상 밖의 수학이었다. 기존의 수학적 방법만으로는 불가능했고, 인공지능 방법을 추가하고 이를 색다르게 활용한 결과였다. 공학적으로는 스타일과 콘텐츠 입력이 동일할 때 발생하는 스타일전이의 오류를 발견했다고 볼 수 있다. 덕분에 오류를 이해하고 제거하기 위한 새로운 연구를 시작할 수 있게 되었다. 더불어 예술적으로는 오류에서 출발한 새로운 표현법을 발견한 것이다. 반가운 오류였고, 즐거운 오류였다. 이러한 오류의 해석은 공학적 관점에서는 새로운 세계였다. 하지만, 이러한

오류를 본격적으로 작품 제작에 활용하기 위해서는 자유자재로 오류의 재현이 가능해야 했다. 결국 ‘오류 정복’의 과정이 필요했다.

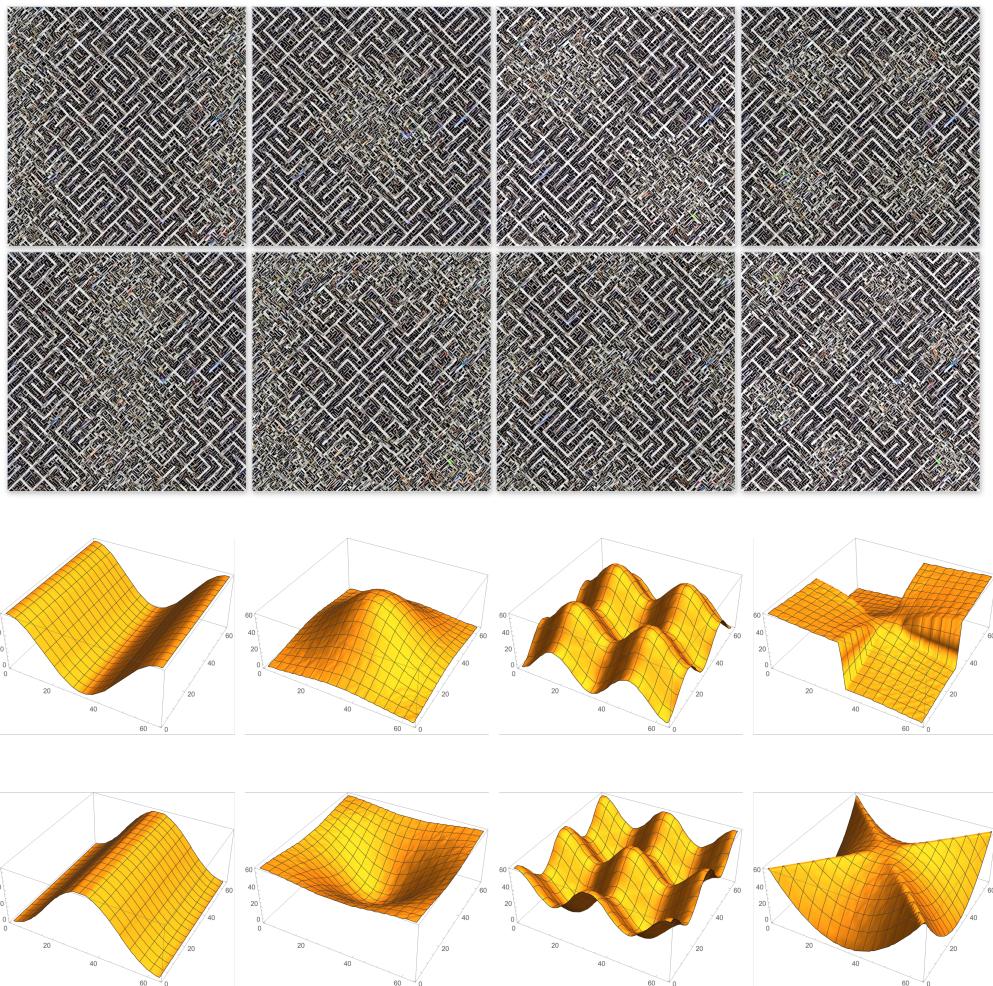


그림 12. 라인그리드 – 무질서의 진화, 그리고 이를 생성한 수학곡면 (Line Grid – Evolution of Disorder, 2019)

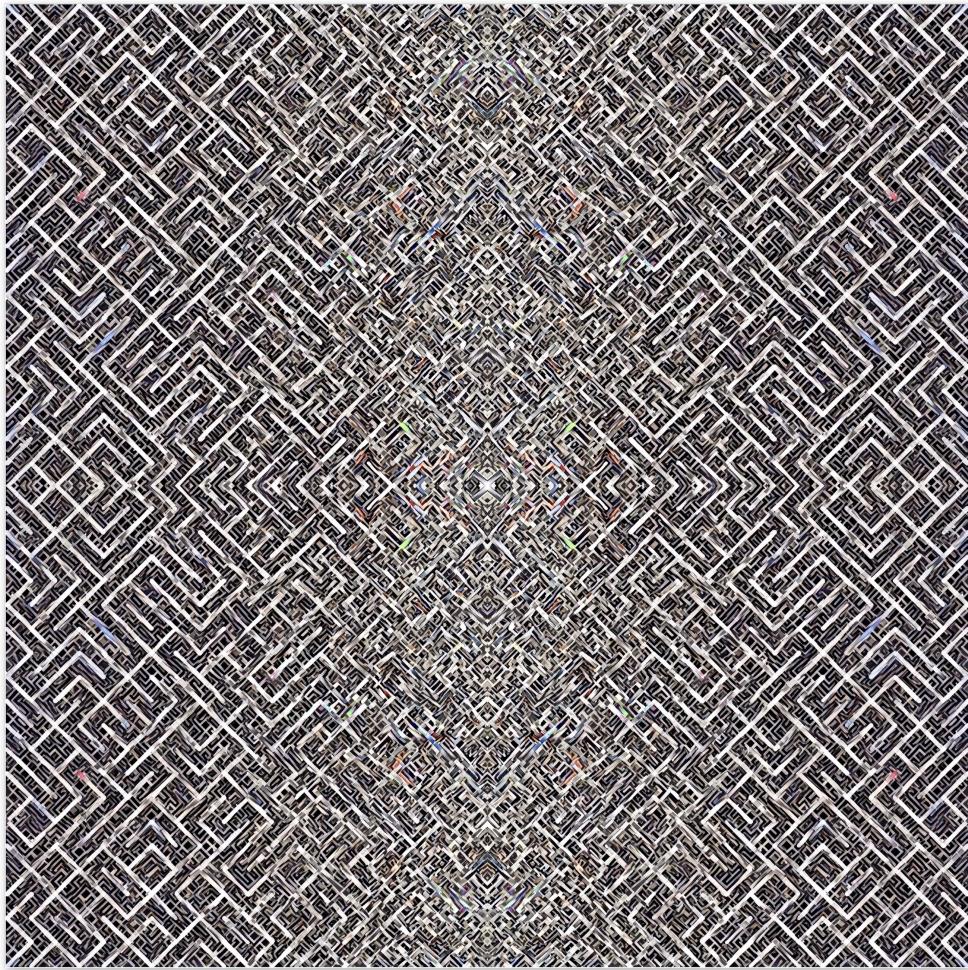


그림 13. 라인그리드 – 모호한 경계 (Line Grid – Ambiguous Boundary, 2019)

—

코드 한 줄로 적어보는 ‘코드로 그린 그림’

‘코드로 그린 그림’ 제작 과정은 한 줄의 코드로 표현이 가능하다. 아래는 이에 해당하는 한 줄의 코드이다¹⁶.

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

컴퓨터 코드를 처음 보는 분들에게는 낯선 외국어처럼 보이겠지만, 그 의미는 복잡하지 않다. 즉, 함수 f 를 $Table$ 로 반복 수행하여 이미지($imgL$)들을 출력하는 과정이다. 조금 더 자세히 살펴보자.

¹⁶ ‘코드로 그린 그림’ 작품 제작에는 Mathematica에서 구동되는 Wolfram 언어를 사용하였다.

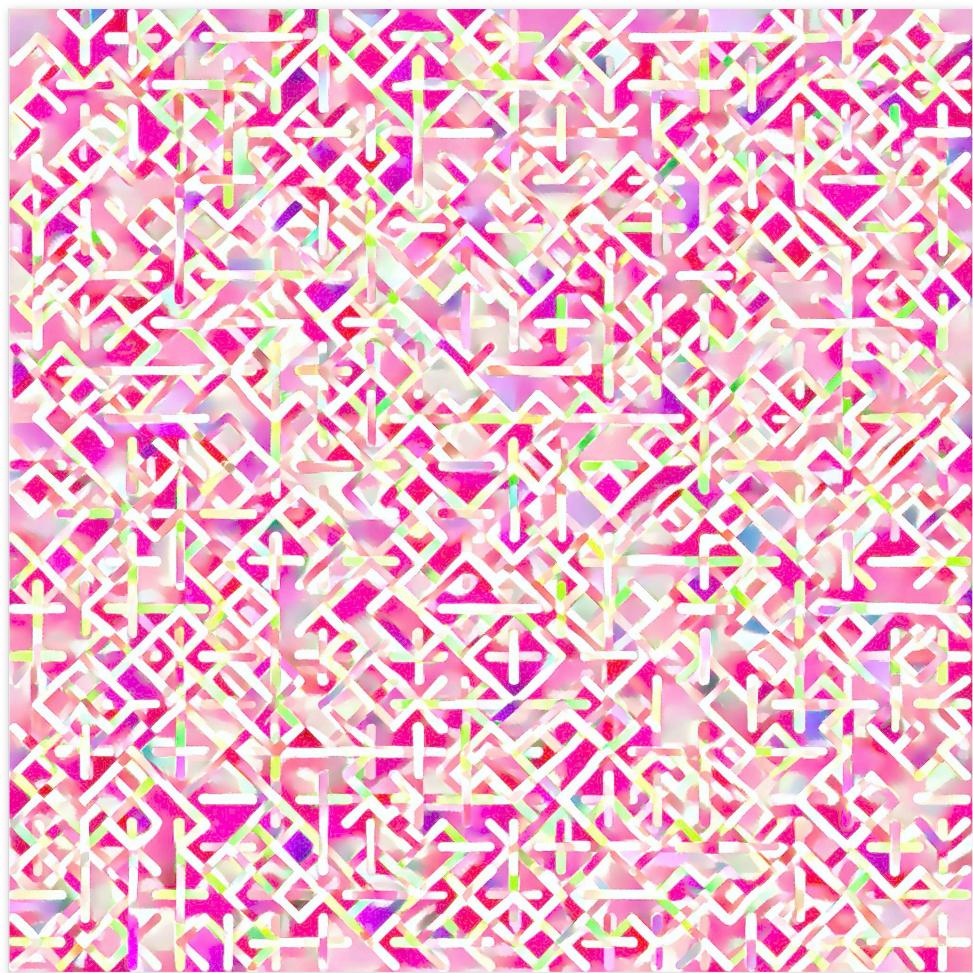


그림 14. 라인그리드 – 봄 (Line Grid – Spring, 2020)

기호 f 는 함수이다. 수학시간에 배운 $f(x)=y$ 에 해당한다. 함수 f 로 그림을 그리기 위해서는 입력이 필요한데, 가장 중요한 입력은 상상(imagination)이다. 이 상상을 감각가능하도록 구현할 함수 f 를 만드는 것이 ‘코드로 그린 그림’의 작가가 해야 하는 가장 중요한 일의 하나이다¹⁷. 상징적인 의미로 작가(Author→“Joo-Haeng Lee”)를 명시하는 것도 함수의 입력이 된다. 다른 중요한 입력은 함수의 동작을 제어하는 인수(Param→ p)이다. 예를 들어, ‘라인 그리드 – 무질서의 진화’에서 패턴의 복잡도를 명시하는 것이 제어 인수에 해당한다.

특정한 함수에 어떤 제어 인수가 필요한지를 결정하는 데는 작가의 지식과 직관이 큰 역할을 한다. 그리고 그 제어 인수의 값으로 가장 적당한 값이 무엇인지를 알아내는 데는 수 많은 반복 실험이 필요하다. 이러한 반복은 Table 구문 안에서 서로 다른 제어 인수값(paramL)들에 대해서 수행된다. 이 과정에서 오류를 발견하기도 하고, 이를 마스터하여 새로운 표현기법으로 삼기도 한다. 서로 다른 인수에 대한 반복 실험의 결과로 여러 이미지들(imgL)을 얻게 된다. 이렇게 얻은 이미지들에서 작품을 선택하는 것도 작가의 역할이다. 전시를 위한 작품을 선택할 수도 있고, 상업적인 작품을 고를 수도 있다. 그리고, 그 선택은 작가마다 다를 것이다.

¹⁷ 함수를 직접 만들 수 없다면, 인공지능이 만든 함수를 활용할 수도 있다.

이 과정은 일반적으로 예술가가 새로운 도구와 기법을 마스터하기 위해 겪어야 하는 실험의 과정과도 같다. 더 나아가 새로운 예술적 상상과 개념을 작품으로 감각할 수 있도록 구현하는 창작의 과정이기도 하다. 이는 피카소가 자신의 작품을 실험이자 연구라고 정의한 것과도 통한다¹⁸.

"내 그림들은 모두 연구이자 실험이다. 나는 예술 작품으로서 그림을 그리지 않는다. 모두가 연구이다. 끊임없이 탐색을 하고, 이 연구에는 논리적인 순서가 존재한다." – 파블로 피카소

"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

'코드로 그린 그림'은 인간의 역할이 필요하지 않은 차가운 과정으로 느껴질 수 있다. 하지만, 위에서 살펴본 바와 같이 '코드로 그린 그림'의 작업 과정을 코드로 표현했을 때, 오히려 여타의 예술에서처럼 인간의 역할이 필수적임을 알게 된다. 상상을 하고, 함수를 작성하고, 인수를 선택하고, 오류를 발견하고, 새로운 기법을 마스터하고, 이미지들에서 작품을 선정하는 여러 과정에서 인간 작가의 역할이 필수적이다.

이제 '코드로 그린 그림'이 '코드가 그린 그림'과는 전혀 다르다는 것을 이해했을 것이다. 코드는 물감, 붓, 캔버스와 같이 작가가 사용하는 도구일 뿐이다. 따라서 '코드로 그린 그림'을 영어로는 'code painting'이라고 한다. 수채화를 의미하는 'watercolor painting'과 같은 맥락이다. 나에게는 전통적인 컴퓨터 그래픽스 코드가 오랜 도구였었다. 지금은 인공지능이 만든 함수를 새로운 도구로 맞이하고 있는 중이다.

최근 인공지능이 작가가 될 수 있는지에 대한 논란이 있다. 나는 '코드로 그린 그림'에서 필요했던 인간의 역할 모두를 인공지능이 대신할 수 있다면, 그때는 인공지능을 작가로 인정할 수 있을 것 같다고 생각한다. 하지만, 이는 꽤나 먼 미래의 이야기가 아닐까 생각한다. 그 전에 인공지능은 예술가의 똑똑한 도구로 발전해 갈 것이다. 그리고, 점점 더 많은 예술가들이 인공지능을 새로운 도구로 받아 들이게 될 것이다. 나는 인공지능의 원리를 규명하고 발전시키는 컴퓨터 과학자로서, 동시에 인공지능을 도구로 활용하는 작가로서, 과학과 예술의 모호한 경계를 탐구하고 그 안에서 즐겨볼까 한다.

¹⁸ 반면에 아인슈타인은 "과학의 본질은 상상력을 이용해서 단순성과 같은 미적 가치를 추구하는 활동이다"라고 하였다. 과학과 예술의 접점은 생각보다 가까이 있는 것 같다. 더 흥미로운 내용이 홍성우님의 다음 논문에 담겨 있다. "과학과 예술: 그 수렴과 접점을 위한 시론"