

The Tools of Generative Art, from Flash to Neural Networks



The artist Sougwen Chung collaborating with D.O.U.G._1 (Drawing Operations Unit: Generation_1) on the performance *Drawing Operations*, 2015. Photo Drew Gurian.

Like it or not, we are all computer nerds now. All aspects of our lives are driven by computation and algorithms: how we learn, work, play, even date. Given this situation, one could argue that generative art—work created at least in part with autonomous, automated systems—is the art that best reflects our time.

Generative art was initially rejected by the cultural establishment as the domain of computer scientists and mathematicians. Grace Hertlein says a colleague called her a “whore” and a “traitor” for her use of the computer as an art-making tool in the late 1960s.¹ In a 1970 *New York Times* review, critic John Canaday compared a display of computer art he saw at a convention to “popular sideshows” and “circuses.”² But recent years have seen a spike in institutional interest in generative art, as evidenced by a number of museum shows.³ Perhaps this embrace is linked to the increased accessibility of technology, as computers and network connections have become commonplace in homes in the last two decades.

Related Articles

These advances have been accompanied by shifts in who can make generative art, how they make it, what it looks like, and even the themes and topics that it is capable of addressing. Because the tools and the work are tightly coupled, the history of generative art can be seen as a history of developments in software and hardware.

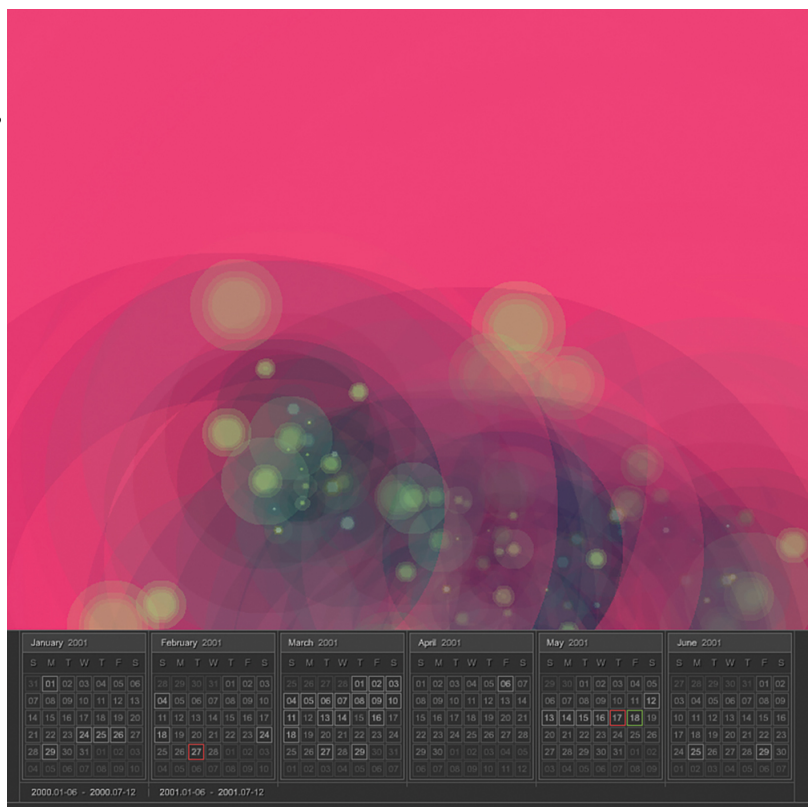
In the '90s, Joshua Davis was an art student at Pratt Institute in Brooklyn, learning to paint by day and absorbing everything he could about programming and building websites by night. Eventually, he asked himself why he was bothering to learn to paint with the same tools and

techniques that had been around for hundreds of years instead of focusing solely on computers and the internet, which had yet to be fully explored by artists. He decided to quit school and make websites full-time. Davis was not the only one. The pull in those days to drop out and join a dot-com start-up as a web designer was strong.

The web was catching on fast, and the demand for great digital content far exceeded the number of people with the skills to produce it. [Flash](#), a tool for creating animations and other multimedia content, was born into this atmosphere of pent-up demand. Initially launched as FutureSplash Animator in 1995 by a small San Diego-based start-up called FutureWave, the software was acquired by competitor Macromedia the following year and rebranded as Flash. Adobe bought it in 2005.

A Flash plug-in enabled multimedia content made with the software to play in browsers. It spread rapidly, and was installed on over 98 percent of networked computers at its peak.⁴ (Other programs have since replaced it, and Adobe is retiring Flash this year.) Most designers used its time-line interface, which closely resembled video editing software, to create simple animations and eye-catching banner ads. But Flash employed a programming language called ActionScript that allowed users to put code directly onto graphics or frames in the time line of an animation. This way, basic

actions like scaling, rotating, opacity changes, and even morphing one shape into another (a process called tweening in Flash) could be accomplished programmatically when the play head hit a particular frame and triggered the code. By



looping these commands and repeatedly drawing shapes onto the screen as you scaled and rotated them, you could use Flash as a generative art tool rather than simply as a linear animation aid.

Joshua Davis: *ps2.praystation.v4*, 2001, website created with Flash software.

Flash helped create a new breed of artist/developer, unschooled in traditional computer science and unafraid to dive in and experiment by sharing code and learning by doing. Many contemporary generative artists cite Davis's site praystation.com as the inspiration for their interest in creative coding. The site, which won the Net Excellence prize at the Ars Electronica festival in Linz, Austria, in 2001, initially served as Davis's personal portfolio. He was flooded with emails inquiring about his pioneering techniques. Overwhelmed and uncertain how to respond to all the

requests, Davis decided to post his project files and his code on his site. Many artists got their start learning directly from Davis's code and, in turn, decided to follow suit and freely share their own.

The Austrian artist who goes by the alias LIA used both an older program called Director, optimized for making multimedia CD-ROMs, and later, the more web-friendly Flash, to program interactive works. LIA's site re-move.org, online in the late 1990s, combined motion, sound, and interactivity, recalling music videos and video games.

Flash's capabilities shaped the look and feel of Davis's and LIA's early work. Before 2006, the program could not manipulate bitmaps, or rows of pixels, so artists had to work with the hard-edge geometry of vector graphics and limited palettes in a small number of on-screen elements. Flash was excellent at drawing and animating flat two-dimensional shapes and text in a web-friendly format, but less so at performing the calculations required for complex simulations that allow for great detail. A minimal look became the signature of generative art in the late '90s.

Flash provided artists with the tools to create basic commands and

Detail of Ben Fry's drawing *All Streets*, 2007, showing the roadless topography of the Appalachian Mountains.

inspired the open sourcing of code within a growing community. But the software itself was expensive.

Programmers typically work in a development environment where they can access the procedures and tools they need to write, test, and debug code. Flash's development environment was nonstandard and limited because of its origins as an animation tool.

To make generative art more accessible and flexible, computer scientist and artist John Maeda started a project called Design by Numbers at the MIT Media Lab with the aim of developing an open-source platform where users could learn transferable programming skills. He recruited several students to help work on it, including Ben Fry and Casey Reas, who eventually built a robust programming language that they would call [Processing](#).

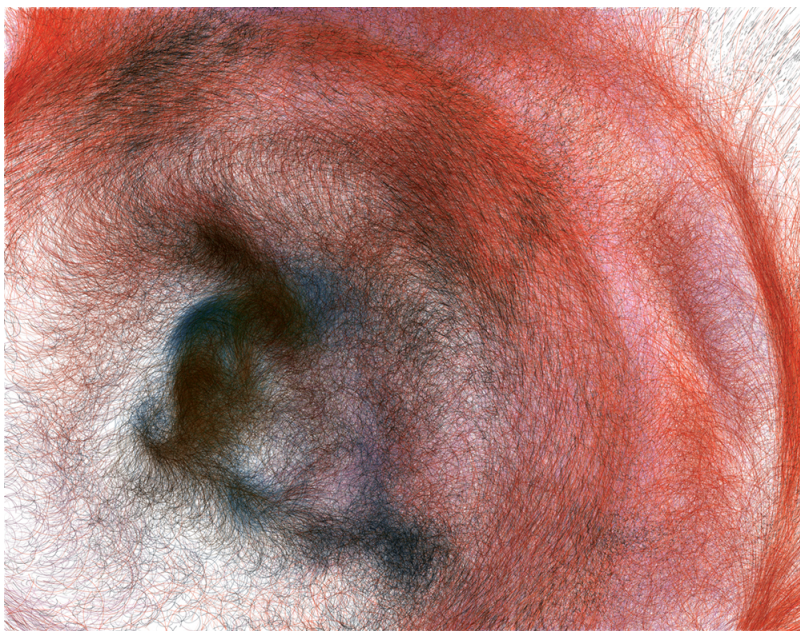
Flash could quickly render simple animations, but Processing could calculate properties of thousands of graphical objects. The difference between them is apparent when you consider their respective architectures. The Flash editor, full of drawing tools, is geared toward graphic designers. Processing's development environment is an empty text window—an interface familiar to a software programmer.

Where Flash was an animation tool that happened to include a scripting language, Processing was built to be fast to write and easy to read, in order to help artists and designers learn to program. It came with sample projects that outlined basic

computer science principles, such as loops (a list of commands set to repeat until a prescribed goal is met), functions (a procedure that returns a specific value), and arrays (a way to store data that makes it easy to retrieve and use). Though you could publish Java applets from Processing and post them to the web, the software was not as browser-friendly as Flash. It was intended as an educational tool, not a commercial one, as Flash had been from the start.

In the early years of Processing, many artists explored concepts based on patterns that occur in nature. Reas was interested in the phenomenon of emergence, a process in which a collective entity, such as a flock of birds or a school of fish, begins to exhibit properties that its individual members do not. His work *MicroImage* (2002) is an animation built from repetitions of relatively simple parts and commands. From thousands of dots, each programmed to react simply to its surroundings, a more complex system takes shape. The result is a gorgeous procedural animation that feels like a living, breathing, pen-and-ink drawing.

Jared Tarbell is another master of generative art based on principles common to nature and computer science. The textures in his digital drawing *Intersection Aggregate* (2004) more closely resemble those of earthy materials like clay, straw, and ash than they do pixels or the vectors associated with early generative art. *Intersection Aggregate* was Tarbell's



Casey Reas: *MicroImage A-03*, 2002, pigment print, 11 by 14 inches. Courtesy DAM Gallery, Berlin.

response to a prompt from Reas, who had been commissioned by the Whitney Museum in 2004 to create a web-based project inspired by the drawing instructions of American conceptual artist Sol LeWitt. Reas devised open-ended instructions to be

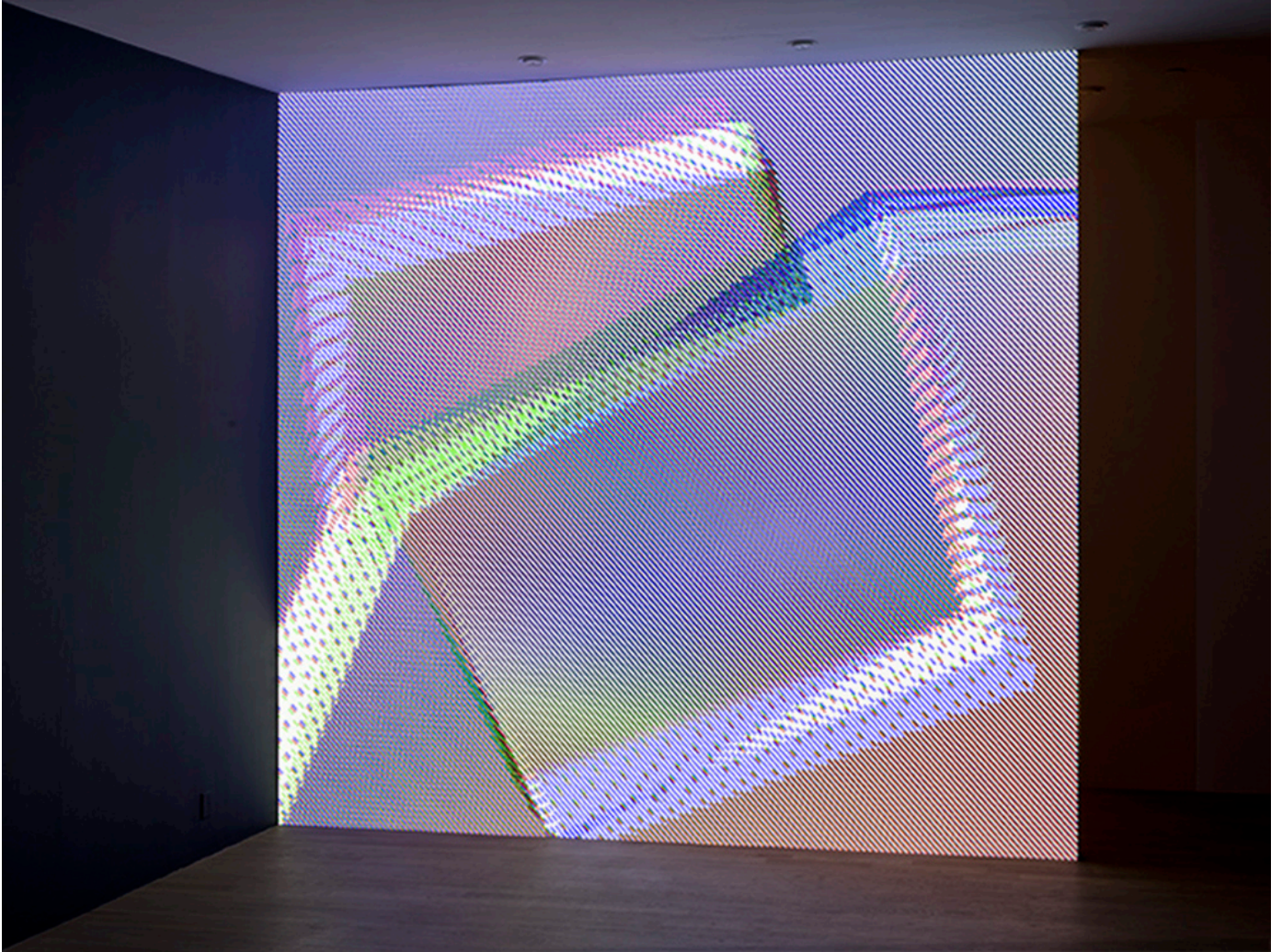
followed by himself and three other artists: “A surface filled with 100 medium to small sized circles. Each circle has a different size and direction, but moves at the same slow rate. Display: A. The instantaneous intersection of the circles. B. The aggregate intersections of the circles.” The genius of Tarbell’s entry is that his interpretation of this mechanical instruction spawns an artwork that seems to have grown from the soil rather than from the computation of ones and zeros.

I see these Reas and Tarbell works of the early 2000s as a breakthrough not only for generative art, but for art history in general. For thousands of years, artists have tried to reveal nature’s essence by copying and imitating its outward manifestations. *MicroImage* and *Intersection Aggregate* go beyond that, deploying code that distills and emulates the

generative forces of the natural world.

If tools like Flash and Processing provided artists with new vehicles for creating art, new fuel arrived in the early 2000s, with an explosion in publicly available data sets. By 2007, the world's capacity to store information had been growing at a rate of 25 percent to 35 percent per year for two decades, and artists were increasingly making use of it.⁵ Ben Fry, Aaron Koblin, Robert Hodgin, and others showed that data visualizations could provoke questions as well as answer the ones that more conventional, practical charts and graphs do. Fry's drawing *All Streets* (2007) depicts 26 million individual road segments from the contiguous United States in black on a cream-colored background. The work can be viewed on Fry's website, but is best appreciated as a large print. It is hard to fathom the level of detail this project involves, but if you spent one minute drawing each segment and drew for twelve hours a day without breaks, it would take you roughly one hundred years to replicate *All Streets* by hand. Fry includes no other geographic markers, but the road segments alone make the image identifiable. Fry evokes an entire country's population and infrastructure in a single image, with a complexity achievable only through the algorithms of an inexhaustible machine. While some might frame Fry's *All Streets* as a classic work of cartography, I see it more as an evolution of landscape painting mixed with generative art. *All Streets* is not a map. It can't be used for

navigation. Instead, it is a rendering that provides an awe-inspiring macro view of our built environment.



Casey Reas: *Still Life (RGB-AV A)*, 2016, software, computer, speakers, and projector, dimensions variable. *Courtesy bitforms gallery, New York. Photo John Berens.*

Since the US government began funding research in artificial intelligence in the 1950s, there have been periods when programs failed to meet the unrealistic and hyped expectations, and the money was cut. During these periods, known as “AI winters,” some research continued at universities and corporations under other names, like

“search algorithms” and “[machine learning](#).” The last winter lasted from 1987 to 1993. It thawed thanks to more affordable computer power and the availability of large data sets. Today AI is popular—though media hype creates a risk of another winter.

Artists’ interest in AI took off after computer scientist Ian Goodfellow released his seminal paper outlining the concept of generative adversarial networks (GANs) in 2014.⁶ He developed a system of two neural networks: a discriminator and a generator. The generator looks at large sets of training data and tries to produce something that resembles the data so closely that the discriminator cannot tell it was produced by another network. The goal is to optimize the system so that the generated output is impossible to distinguish from the real inputs.

Mario Klingemann’s *Memories of Passersby I* (2018) is an example of the use of GANs as a tool for making art. Klingemann devised a system that creates an endless series of eerie portraits based on training images of historical paintings. The portraits are displayed on two screens positioned side by side above a chestnut console that houses the work’s hardware. The location of the screens relative to the console makes them look almost like mirrors, so that the portraits feel all the more intimate, confrontational, and disturbing.

The figures in Klingemann's work are pale and gaunt, resembling those in old photos of asylum patients or medical catalogues documenting human deformities. The faces enter the world only briefly, but they all have the look of old souls



Mario Klingemann: *Memories of Passersby I*, 2018, console, computer, and two monitors.

carrying the weight of a troubled past. This haunted aesthetic is Klingemann's hallmark; he has always avoided the tendency to make digital art with a polished and shiny aesthetic. GANs trained on photos tend to introduce bizarre quirks as they struggle to produce something like the input images, and Klingemann relishes the results. They are quite different from generative art that uses iterative commands to draw vector-based shapes to the screen.

Klingemann's interactive website *Fractal Machines* (2001), made with Flash, comprises two-dimensional vectors animated procedurally to move, rotate, and scale. The piece combines images of fanciful gears and other mechanistic components that interlock and form machines or robots. The colors are muted grays. The work looks like the dream of a

mad scientist from a past century. To create it, Klingemann had to enter code to generate the shapes, choreograph their motion, and set the rules according to which they interact. The artist can make adjustments to this environment, including the introduction of randomness. By contrast, each face produced by *Memories of Passersby I* is as much a surprise to Klingemann as it is to us. While he has control over the training of the GAN and can manipulate parameters to influence the general feel of the output, Klingemann relinquishes the authority to select which portraits viewers see. They emerge directly from the machine to the audience.

Until recently, creating work with GANs and other machine learning models required deep technical knowledge and hard-won programming skills like those developed by the self-taught Klingemann. But this barrier is starting to fall as new tools like Artbreeder and Runway ML make GANs and other machine learning models more accessible. In fact, the amount of skill required to generate images using GANs is now so low—much lower than using Flash and Processing—that we are seeing a flood of images with the GAN aesthetic of slightly deformed and creepy faces.

Artbreeder, a tool created in 2018 by artist Joel Simon, has already produced more than 54 million images. Simon radically simplified the process of making art with GANs. Users of the tool just click on images to “breed” them,



Inkjet print by Helena Sarin, 2019.

dragging sliders back and forth to increase or decrease the amount of influence that the source images have on the output.

Runway ML is a more advanced program. Born of an academic research

project by cofounder Cristóbal Valenzuela, the software has been designed to accelerate the movement of new algorithms and models from research to software. Runway's intuitive interface makes it easy to get started quickly, but it also allows for fine-tuning and control over multiple machine learning models, including ones that colorize black-and-white photos, transfer style from one image to another, recognize faces, and turn doodles into photorealistic images, among many others. You can build sophisticated pipelines that sequence various models and shorten execution time by connecting to the cloud. This means Runway offers more flexibility and power than a tool like Artbreeder, which trades complexity for ease of use.

But even work created in Runway can seem homogeneous, as much of it features the surreal quality common to GAN art, or some kind of style transfer (e.g., a painting that has

been made to look like a photograph). The arbitrary execution and application of these Runway models to random images does not create art, at least not very interesting art. The more the effects are used by more people, the less artistic they feel.

We experienced something similar to this algorithmic overkill in the mid-1990s, when Photoshop filters first became available and were used widely without artful manipulation. They quickly became kitsch, and died out. Artist-friendly tools have democratized creative coding over the last two decades. But as barriers to creating generative art fall, artists must find new ways to differentiate themselves or get lost in the seemingly infinite and repetitive imagery being produced.



Perhaps the best way to establish a singular vision is to increase the emphasis of human creativity in part by reintroducing elements of analog art production. Sougwen Chung, Anna Ridler, and Helena Sarin are some of the artists who train GANs on their

own drawings and

paintings—bodies of visual information that are distinctly theirs, unlike large public data sets. Sarin has also used visuals generated by AI as the basis for works made with various analog processes, from glass fusing and pottery to monotypes and screen-printing. By combining these physical art-making methods with cutting-edge digital techniques, Sarin has developed her own language that is warmer and more physically engaging than push-button GAN images.

Like Sarin, Sougwen Chung trained a neural network on her own drawings and employs analog art processes to make her final work. But she has also programmed robots to draw alongside her on large canvases on the floor in a series of live performances. Chung stresses that even during the programming of the machine learning model, her own hand is present. Though the machines draw in the style of her work, she is still surprised by some of the specific marks the machines make, and she incorporates these new forms in her own drawings. The results are large, looping, sinuous structures that fold together the impulses of the artist and her robot collaborators.

Many artistic processes can be described as algorithmic. Artists follow sequences or steps in the production of their own work. Often what makes one artist's practice more interesting than another's is spontaneity, a willingness to

challenge their own system. Sometimes, it is computerized tools that make this leap possible.

Our future is an exciting one that combines analog and digital, human and machine, rather than elevating one at the expense of the other. If the history of generative art is our generation's history, let it be told through the narrative of the talented artists who embraced the new ideas and tools of our time.

¹ Grace Hertlein, quoted in Grant D. Taylor, *When the Machine Made Art: The Troubled History of Computer Art*, London, Bloomsbury, 2014, p. 7.

² John Canaday, "Less Art, More Computer, Please," *New York Times*, Aug. 30, 1970, p. 87, quoted in Taylor, p. 6.

³ These include "Artists & Robots," at the Grand Palais, Paris, Apr. 5–July 9, 2018; "Chance and Control: Art in the Age of Computers" at the Victoria and Albert Museum, London, July 7–Nov. 18, 2018; "Programmed: Rules, Codes, and Choreographies in Art, 1965–2018," at the Whitney Museum of American Art, Sept. 28, 2018–Apr. 14, 2019; "Coding the World" at the Centre Pompidou, Paris, June 14–Aug. 27, 2019; and "Face Values: Exploring Artificial Intelligence" at the Cooper-Hewitt Museum, New York, through May 17, 2020.

⁴ Rafia Shaikh, “Get Ready to Say Farewell to Flash Player as Adobe Has Finally Decided to Pull the Plug,” wccftech, July 25, 2017, wccftech.com.

⁵ The growth rate is estimated by Martin Hilbert, who describes his method in “How much information is there in the ‘information society’?,” a blog post for the Royal Statistical Society, rss.onlinelibrary.wiley.com.

⁶ Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, “Generative Adversarial Nets,” June 10, 2014, arxiv.org.

This article appears under the title “Doing Things with Computers” in the January 2020 issue, pp. 34–41.