

Notes on “Phototaxis”



Casey REAS · Follow

8 min read · Sep 19, 2021



129



I started learning to code in 1999 to create drawings that I couldn't make with my hands. My original way into the visual arts was through drawing and my way into coding was through a new way of thinking about drawing. I was pursuing the mysteries of “emergence” as explored by artificial life researchers and I was applying these ideas to making pictures.

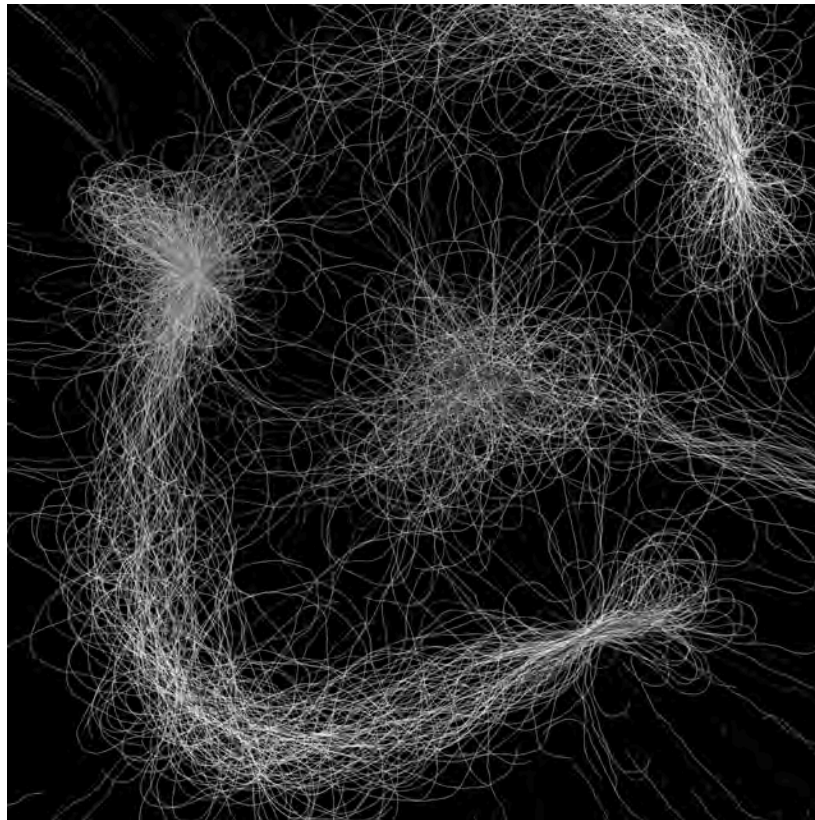
Late in my time studying Media Arts and Sciences as a graduate student at MIT, I found my way into an Artificial Intelligence class taught by Rodney Brooks. In that class, I was introduced to the book *Vehicles* written by the neuroanatomist Valentino Braitenberg. The ideas in this short, but profound book, were the origin for the *Path* and *Tissue* prints that I created 2001–03. Braitenberg's “vehicles” are minimal, simulated bodies that have behaviors similar to living *organisms*. Through writing code, I interpreted these organisms as software machines for drawing.



Still image from Phototaxis, minted on the Ropsten Test Network

The *Path* prints were the first works I exhibited in a gallery as a solo artist. They were shown at the inaugural group show at the bitforms gallery in New York in fall 2001. I continued with the *Tissue* software and prints that were exhibited within a two-person show at bitforms in winter 2002.

Each line in these drawings inscribe the history of one software organism; its coordinates at each state of the simulation are connected together. One color is used to represent each of the four different types of simulated organisms. The color decreases in brightness when the organism move slowly to map high- and low-energy areas. The black-and-white versions of the drawings compress the differences between the organisms into a unified surface.



Still image from Phototaxis, minted on the Ropsten Test Network

A range of software and printed version of this code emerged over a few years of exploration. Prints created from the *Path* and *Tissue* software are currently in the permanent collections at the Victoria and Albert Museum in London, the Frac Centre in France, the Mary and Leigh Block Museum of Art in Chicago, and a number of private collections.

Phototaxis will be released on Art Blocks on 21 September 2021. It's the result of further work on the original source code for the *Path* and *Tissue* prints and software. The original code was written in C++ and was later ported to Processing (Java). In 2021, the Processing code was further developed over a few months and then ported to p5.js (JavaScript) to run through a web browser. The core of the 2001 code survives intact, but the parameters around it have been further refined as “long-form” generative art. The

edition of 1000 to be released on Art Blocks are a narrow set of the potential variations, but they will likely map out the space of possibilities well.

Technical Notes on *Phototaxis*:

- The software halts the simulation after 1000 iterations. The thumbnail image for each mint represents the software in that state. Press ‘P’ to continue the simulation and to further pause and resume it.
- The location of the lights (simulated environmental stimuli) is fixed for each mint and there are always between 2 and 7 lights. Press ‘L’ to see the locations of these lights.
- Press ‘B’ to start the simulation from the beginning.
- Press keys ‘1’ to ‘5’ to see the simulation at different scales

— Casey Reas, 18 September 2021

The following text is an excerpt from “Tissue,” published in CODE: The Language of Our Time. Hatje Cantz Verlag. 2003. Some of the behaviors in Phototaxis on Art Blocks are different from what is described below for the original 2002 Tissue software.

Tissue

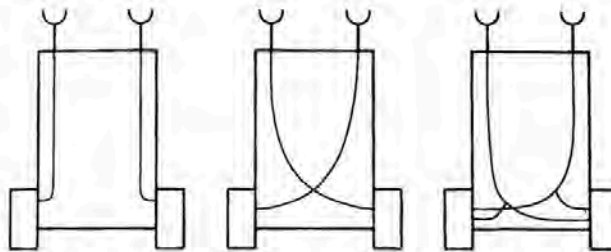
Tissue exposes the movements of autonomous software machines. Each line in the image reveals the path of each machine’s movement as it responds to stimuli in its environment. People interact with the software by positioning the stimuli on the screen. Through exploring different positions, an understanding of the total system emerges from the subtle relations between the simple input and the resulting fluid visual output.

The concept for *Tissue* originated with the book *Vehicles, Experiments in Synthetic Psychology* by Valentino Braitenberg, a neuroanatomist. As Braitenberg spent his career counting fibers in the visual ganglia of flies and synapses in the cerebral cortex of mice, he began to distill generalizations about how nervous systems work. He wrote, “I have been dealing for many years with certain structures within animal brains that seemed to be interpretable as pieces of computing machinery because of their simplicity and/or regularity.” He uses the term synthetic psychology to define the construction of personality/behavior through building up structures of computing machinery. In *Vehicles*, Braitenberg defines a series of thirteen conceptual constructions by gradually building more complex behavior with the addition of more machinery.



Tissue software and prints at the bitforms gallery in New York in 2002. The software was run from a PC tower attached to a CRT screen, and both were embedded in a custom wood case fabricated in Italy. Three brass pegs were moved by the audience to create the environment for the simulated software machines.

The software machines in *Tissue* are analogous to Braitenberg's Vehicles 2 and 3. Simple layers of code combine to create the deceptively complicated behavior of these machines. Each machine has two software sensors to detect stimuli in the environment and two software actuators to propel itself. The relationships between the sensors and actuators determine the specific behavior for each machine. In the thousands of software machines simultaneously running in *Tissue*, there are four distinct types, each specified with a different color. Each machine has a variable speed, direction, and position of movement, and all machines share the same size, turning rate, and maximum speed.



Examples of machines from Valentino Braitenberg's book *Vehicles*. The semicircles are sensors that are connected to actuators that produce motion. Different wiring patterns create different behaviors.

Each machine continually alters its direction and speed by analyzing its position in relation to the environment. First, it determines the distance from each of its sensors to the first stimulus point:

```

dx = stimulusX — leftSensorX;
dy = stimulusY — leftSensorY;
leftDistance = sqrt(dx*dx + dy*dy);

```

```
dx = stimulusX - rightSensorX;  
dy = stimulusY - rightSensorY;  
rightDistance = sqrt(dx*dx + dy*dy);
```

The software then normalizes the two distance measurements, “leftDistance” and “rightDistance,” and compresses the values between 0.0 and 1.0. The resulting values are input into a nonlinear function, returned, and saved. These two new values are averaged and stored into a variable called “normSensorAverage”:

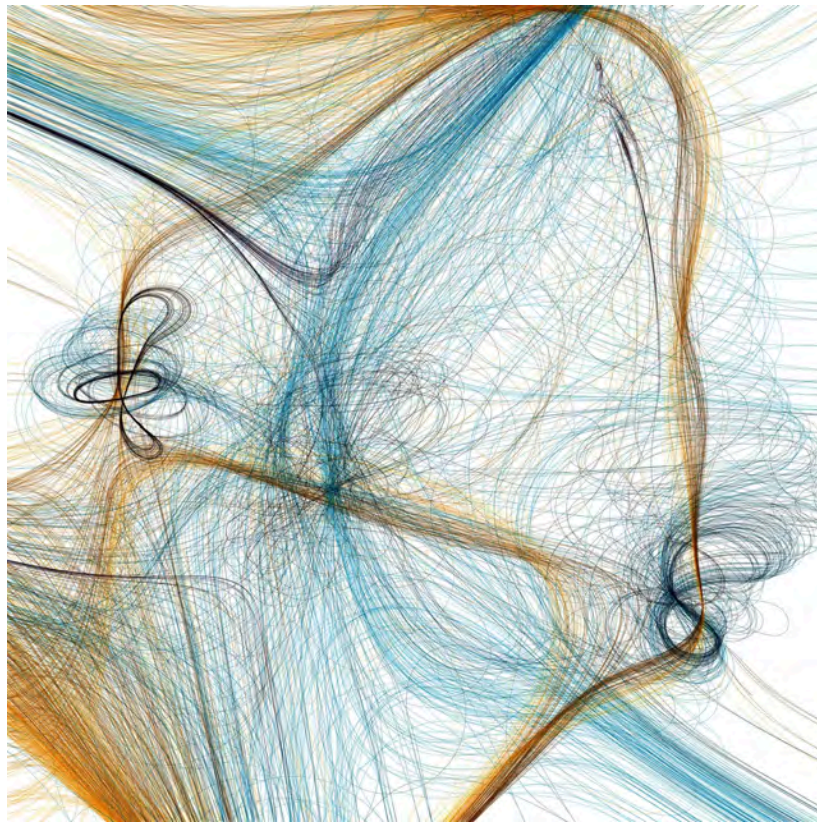
```
normSensorLeft = leftDistance/maxDistance;  
normSensorRight = rightDistance/maxDistance;  
normSensorLeft = hump(normSensorLeft);  
normSensorRight = hump(normSensorRight);  
normSensorAverage = (normSensorLeft + normSensorRight) / 2.0;
```

The speed of the machine is then modified using the value of “normSensorAverage” and the constant value “maxSpeed,” which is the maximum speed that a machine is capable of reaching.

The following text is an excerpt from “Beyond Code,” published in Network Practices: New Strategies in Architecture and Design. Edited by Anthony Burke and Therese Tierney. Princeton Architectural Press, 2007.

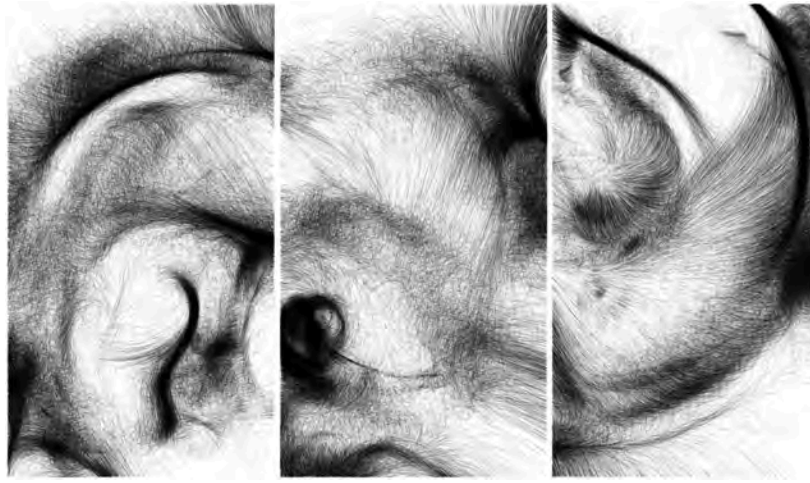
Perspectives on a Process

MicroImage is a series of software, prints, and animations developed from 2001–2003. The ideas for *MicroImage* originated with the book *Vehicles, Experiments in Synthetic Psychology* by the neuroanatomist Valentino Braitenberg.



Path 00 print exhibited at the bitforms gallery in New York in November 2001. 32 × 32 inches.

I first approached visualizing Braitenberg's machines through software. Transferring their behavior from diagrams to computer code required making many decisions about the precise distance between each sensor, the speed of each vehicle, and if there should be any perturbations on their simulated surface. After many preliminary studies, I decided to represent each vehicle as a thin graphic line connecting each vehicle's most recent locations (coordinates). Each segment of the line connects a previous location of the vehicle to a more recent location to visualize its speed and degrees of curvature. The motion of each vehicle type is controlled by changes to the simulated environment. Small changes in the positions of the stimuli create large global changes in the movements of the vehicles. If there are no stimuli, each vehicle moves in a straight line, but as each new point is added, the vehicles respond according to their simulated wiring. The software offers a near infinite set of images, each dependent on the configuration of points that comprise the environment.



MicroImage B-00, B-01, and B-02 prints from 2004. 34 × 24 inches each.

Because the software implementation of *MicroImage* is continually in motion, it is impossible to look at the precise forms that emerge from the interactions between the vehicles and their environment. In addition, because the resolution of computer monitors are so low, it's difficult to precisely see how the lines are positioned in space. I began making static images from the *MicroImage* software to have a different perspective on the system. To make these images at a higher resolution than the default screen, I began exporting each line as a series of vector line segments, which allowed them to be printed at resolutions up to 2880 dots per inch. Over time, I modified the software by changing the parameters of the vehicles' behavior, modifying the line length, and changing the number of vehicles. I challenged some of the assumptions in the original software by changing the way the sensors were related to the motors by testing a series of different nonlinear relationships. Over time, five distinct software variations were produced to show a range of ways to interpret Braitenberg's original text. Through working with the software over a period of months, I exhausted the range of potential images created through each variation and I made eight prints from each software system to show the variation in each. The prints were made while working actively with the software by changing the environment and then exporting specific frames as geometry. Looking at these prints displayed on a wall gives a distinctly different view of the software than can be gleaned from working with the realtime, interactive version.



Tissue mural created for the Rethinking Typologies exhibition at the Art Institute of Chicago in 2012.

These various interpretations of the *MicroImage* process led me to think more about representation. As I claim the core of my work is creating processes and not images, I need to think carefully about how the process is represented. While I'm principally interested in the construction of the process, they must be made visible to be experienced. I made some assumptions in the initial software, print, and animated interpretations and I started to consider these assumptions. I chose to work only with the minimal representation of the line to hopefully expose the process without imposing extraneous visual decoration.



Written by Casey REAS

289 Followers · 52 Following

Follow

Artist and educator. UCLA Department of Design Media Arts. Co-founder of Processing, Processing Foundation. Twitter: @reas