FORM FROM BEHAVIOR

MAINTAINING DESIGN INTENT IN MULTI-AGENT SYSTEMS

Nils Claesson
Chalmers School of Architecture
Department of Architecture and Civil Engineering
Examiner: Daniel Norell
Supervisors: Jonas Lundberg and Kengo Skorick

Nils Claesson
Form From Behavior

Chalmers University of Technology
Department of Architecture and Civil Engineering

Master program of Architecture and Urban Design
Master Thesis | Spring 2018

Supervisor: Kengo Skorick, Jonas Lundberg
Examiner: Daniel Norell

Contents

1
ABSTRACT

This thesis investigates multi-agent systems and generative design with a focus on how these techniques impact design processes. Agents negotiate between different objectives on a micro scale as does the complete system on a macro level. Compared to a traditional architectural workflow with top down decisions the designer here is the creator of a process with control over the relational aspects between the agents and their response to various stimuli. The workflow of algorithmic simulation is supported by intuitive decisions including user control and the possibility of importing geometry from other programs.

Design intent in bottom-up systems is investigated through the technique of stigmergy and is implemented in an urban study where the patterns are used for the analysis of paths. Stigmergy was also used in a facade proposal in which agent movement created a structural system, surface articulation and varying levels of transparency through particles.

A design process that combines top-down and bottom-up methods has been achieved by the the agents responding to generated and imported geometry in a way which still leaves room for emergent behaviour to occur. Generative systems is tightly linked to the degree of control the designer wields over the system. Control was achieved in two ways, through relative parameters controlling agents and through varying resolution of imported geometry.

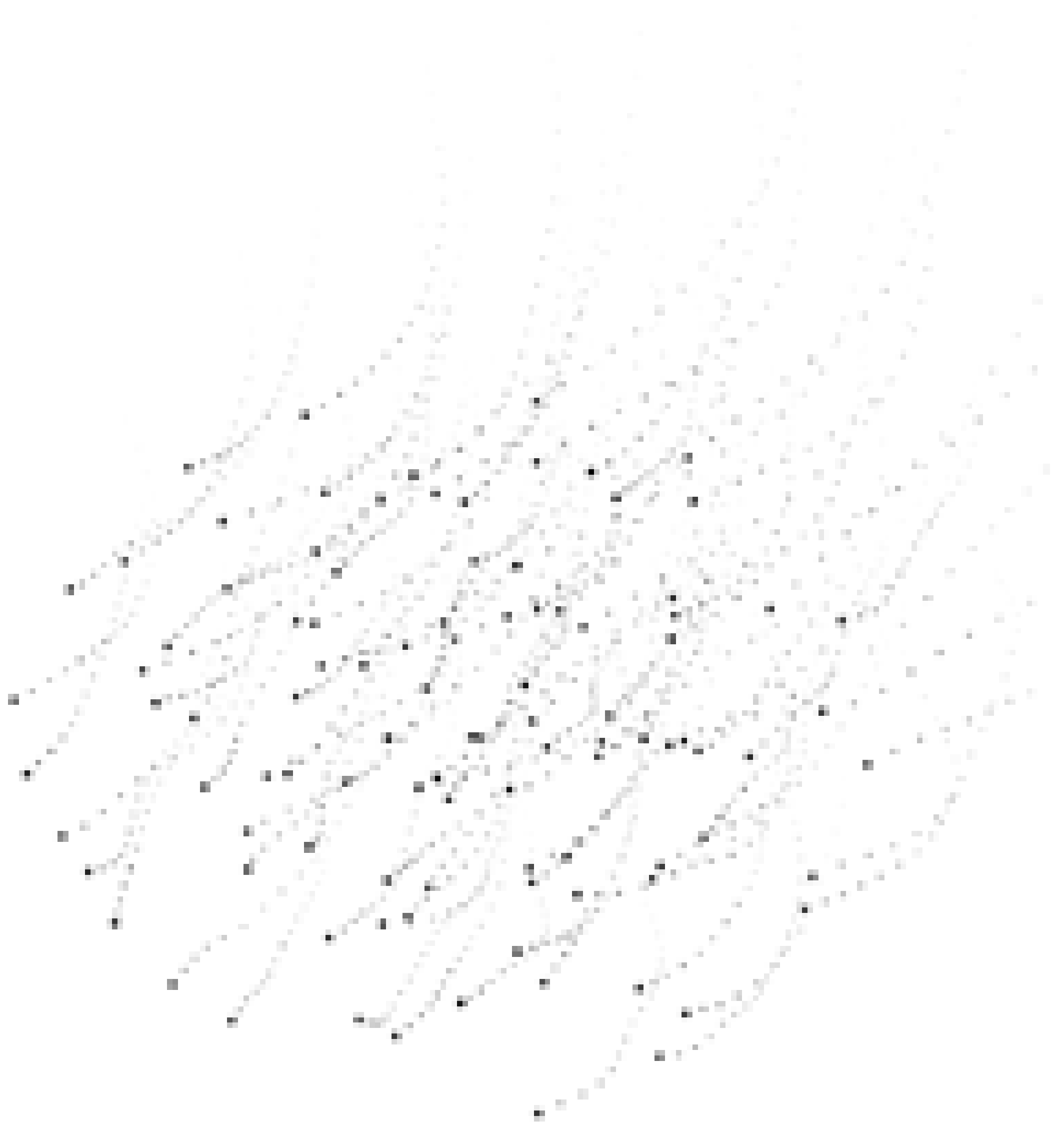
2
STUDENT BACKGROUND

Bsc of Architecture
Chalmers University of Technology

Master's Program: Architecture and Urban Design
Chalmers University of Technology

Studios:
Material and Detail
Design & Communication Tools
Spatial Morphology Studio
Architectural Competitions
Sustainable Building Competition
Building Design Lab

Contact details
Name: Nils Claesson
E-mail: nils.claesson@gmail.com
Phone: +46700809810



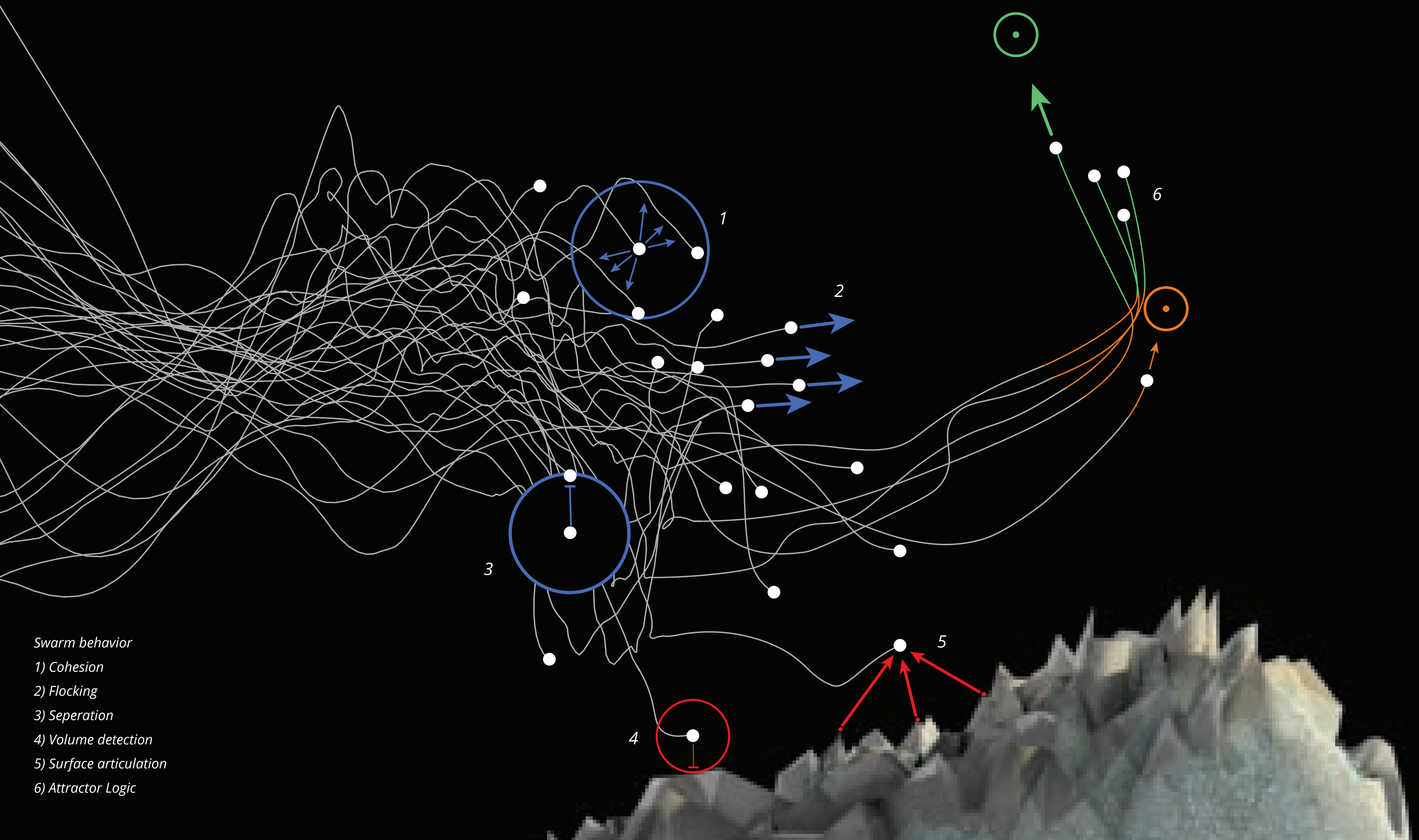
Swarm flocking behavior

3
INTRODUCTION

Multi-agent systems is a type of generative design that is based upon simulations shaped by a set of rules. A close parallell is nature itself that through countless iterations created the world we now live in. Evolution make billions upon billions of computational experiments which continosly redesigns and transforms species. This process is based upon an immense number of individual agents that together creates something that is larger than the sum of its parts. Like the cells that make up the body or atoms who self-organizes to form a tree.

Generative design as well often takes inspiration from nature and is open ended. One cannot with absolute certainty tell what the end result might be, each simulation will produce something new. This variation in output creates an complicated relation to design intent. The scope of possible outcomes need to be reined in to not produce too wild or nonsensical results. A challenge lies in setting up the system itself as one set of rules creates life and another death.

Multi-agent systems is a method that replicates the behavior of bird flocks and schools of fish. The system is composed of individual intelligent agents that inhabit an environment. Each agent has a limited understanding of its surroundings and only reacts to events in its immediate vicinity.



Swarm behavior
1) Cohesion
2) Flocking
3) Seperation
4) Volume detection
5) Surface articulation
6) Attractor Logic

4
PURPOSE AND EXPLORATION

The purpose of this thesis is to examine multi-agent systems and how they can be used in the architectural design process. These tools bring with them a different kind of workflow compared to a regular top-down design process, this change is primarily in how the designer exerts control over the process. The role of the architect has been in continous flux ever since the introduction of digital tools and techniques and it is not stopping now. This thesis is an attempt to understand how this subset of generative digital tools impacts the architectural design process.

5
MAIN QUESTIONS AND OBJECTIVES

How can design intent take effect when working with agent based design processes?

With the difference in how the designer exert control, how is the design process altered?

6
BACKGROUND

DESIGN INTENT AND DIGITAL TECHNIQUE

Greg Lynn posits that there is a right way and a wrong way to use digital techniques or any technique for that matter. He maintains it is important to have a design intent and a goal in mind when using digital tools. There is a danger of giving up control to the computer and end up with results that while interesting lack concept and intent.

I can definitely feel the lure of handing over creative control to the software when working with tools new to me which produces results I haven't seen before. It is important to step back a bit and think of what it is you have produced and how this relates to your vision.

Designing through complex systems, in particular through multi-agent design methodologies, does not represent a loss of control in the design process, however the nature of design and authorship changes. It is a shift from invention of form to the orchestration of processes.
- Roland Snooks [2]

The question of how to maintain design intent while using digital techniques takes on a special twist when using agent-based tools. Swarm systems works through the bottom-up interaction of the individual agents and their response to stimuli, the architect orchestrates a process which in turn creates form. Roland Snooks maintains that the design intent takes place on the level of individual agents where their behavior is manipulated to produce a result in line with the vision of the designer. The topology and the exact dimensions of the output are not directly controlled however the focus shifts towards the organization and interaction of the system (Snooks, 2010).

I posit that design intent cannot be completely in the hands of the bottom-up system. Shifting the focus from topology and dimensions towards organization is not sufficient when applying the technique to a demanding context and program. What is needed in addition to controlling agent behavior is to interpret the output and alter as needed.

Left: Kokkugia - Fibrous House, a project that explores swarms and materiality

[2] Interview with Roland Snooks from thefunambulist.net

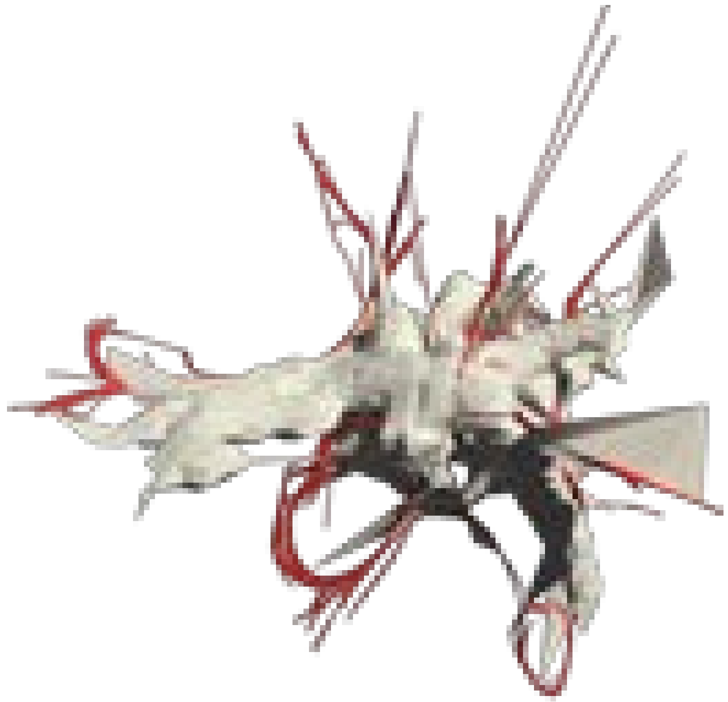
MAINTAINING INTENT

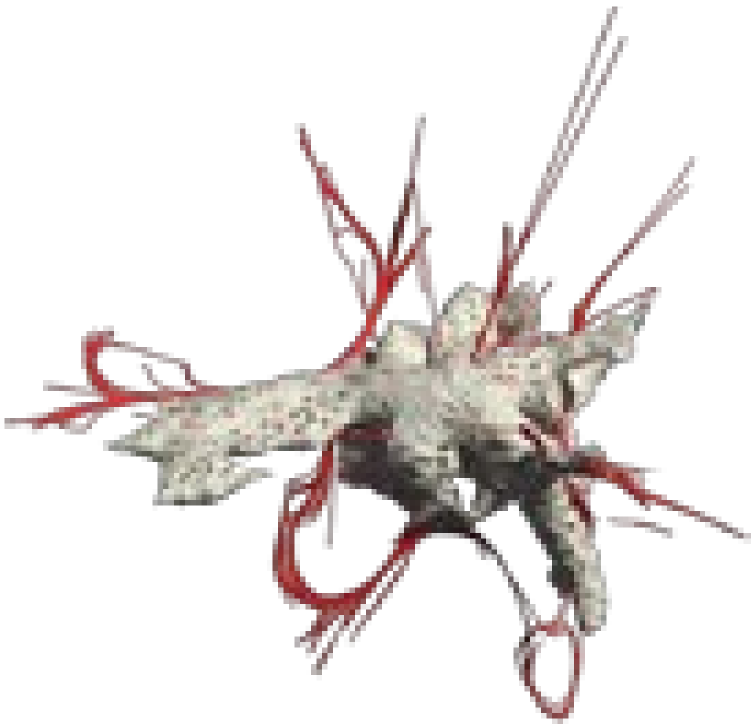
To achieve ones design intent while using swarms one has to imbue it within the algorithmic process and think about how it can relate to architectural values and not simply remain as a form finding exercise. Roland Snooks proposes a method in four steps that approaches this in an iterative way. The first step uses the self-organizing behavior of the swarm to produce emergent form and organization. Input from biology such as ant trails or bird flocks can be used to start off. The output is evaluated for interesting forms and patterns which can be made stronger through the next iteration. The next step is much like the previous one but with an updated code.

The evaluation criteria is changed to look for architectural qualities in the emergent output. Each iteration is tested against the architectural objectives of the project and the behavior of the agents are updated accordingly. The experiment in the picture above belongs to the first step, one might choose to explore the cavernous areas present in the image to the right and rewrite the code to strengthen this.

The third step involves changing the agent behavior to produce global architectural results. An example of this is could be agents that self-organize into surfaces which changes how the forms are percieved on a global level. The outcomes are tested to see if they exhibit any of the global architectural outcomes that are desirable and

Top: Volume arising from agents, same simulation over a timespan of 1 minute





the code is updated and a new iteration begins. The fourth step has to do with the local level and values such as tectonics, ornament, program and structure. This category exhibit emergent behavior on the global as well as on the local scale and the output is tested against both (Snooks, 2014, 57).

Roland Snooks structured method of encoding design intent seems to include all four steps in a single script which could cause problems. Another way to work can be to device specialized scripts that controls the form on different scales. Articulation on the local level could be handled by its own algorithm to allow for more control and less interference on the global level. The notion of mixing the bottom-up processes of agents and top-down choices of the designer is an interesting one. This can be achieved by building in more user interaction within the simulation such as the possibility of selecting certain areas for surface articulation. In addition one needs to see scripting as one design tool among many, and communication between Processing and other programs brings new possibilities and more design space. Having several scripts working on the same project, adding user control and using scripting in conjunction with other programs provides a beneficial counterpoint to the agents emergent behavior.

Top: Left volume directly from Processing, right volume subdivided with Weaverbird afterwards

"[Advanced Computation] has reached a kind of fervor, where technique is promoted over outcomes and effect [...] you lose too much information when everything in an architectural problem has to be processed through an algorithm. Inputs are forced to become quantative or otherwise abstract in order to be computed, so it is not surprising that outputs are also anemic." [2]

7
METHOD

In this project I chose to work with Java programming in the Processing shell. Writing my own scripts in Processing gives me much more freedom and possibility to run simulations due to the effectiveness of the Java environment. Scripting agent behavior is essentially an bottom-up process which I temper with top-down intuitive decisions. For example a volume created by regular modeling tools outside of processing can be imported to be manipulated by agents, then the result is exported to be used by other programs in an intuitive manner.

I promote the use of several programs to achieve the effects I want and to steer clear of the idea that everything needs to be coded and built from algorithms.

[2] Quote from Tom Wiscombe. 'Interview with Olaf Winkler' in build urban architecture and design, March 2010

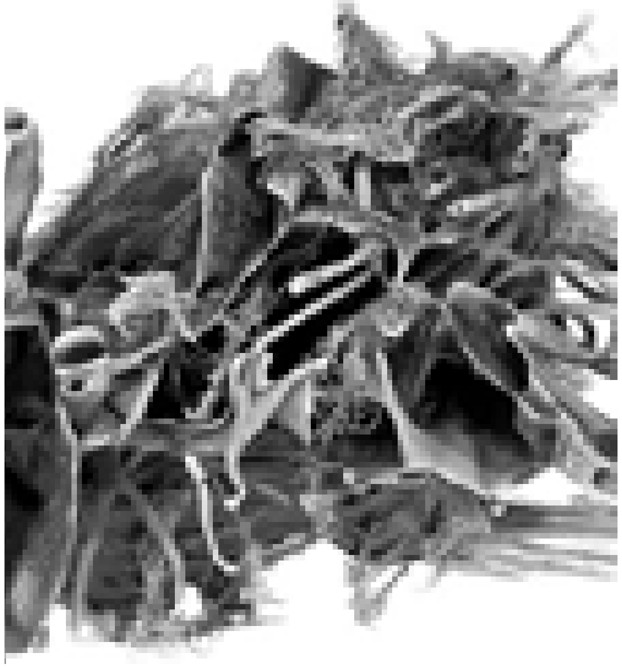
8
REFERENCES

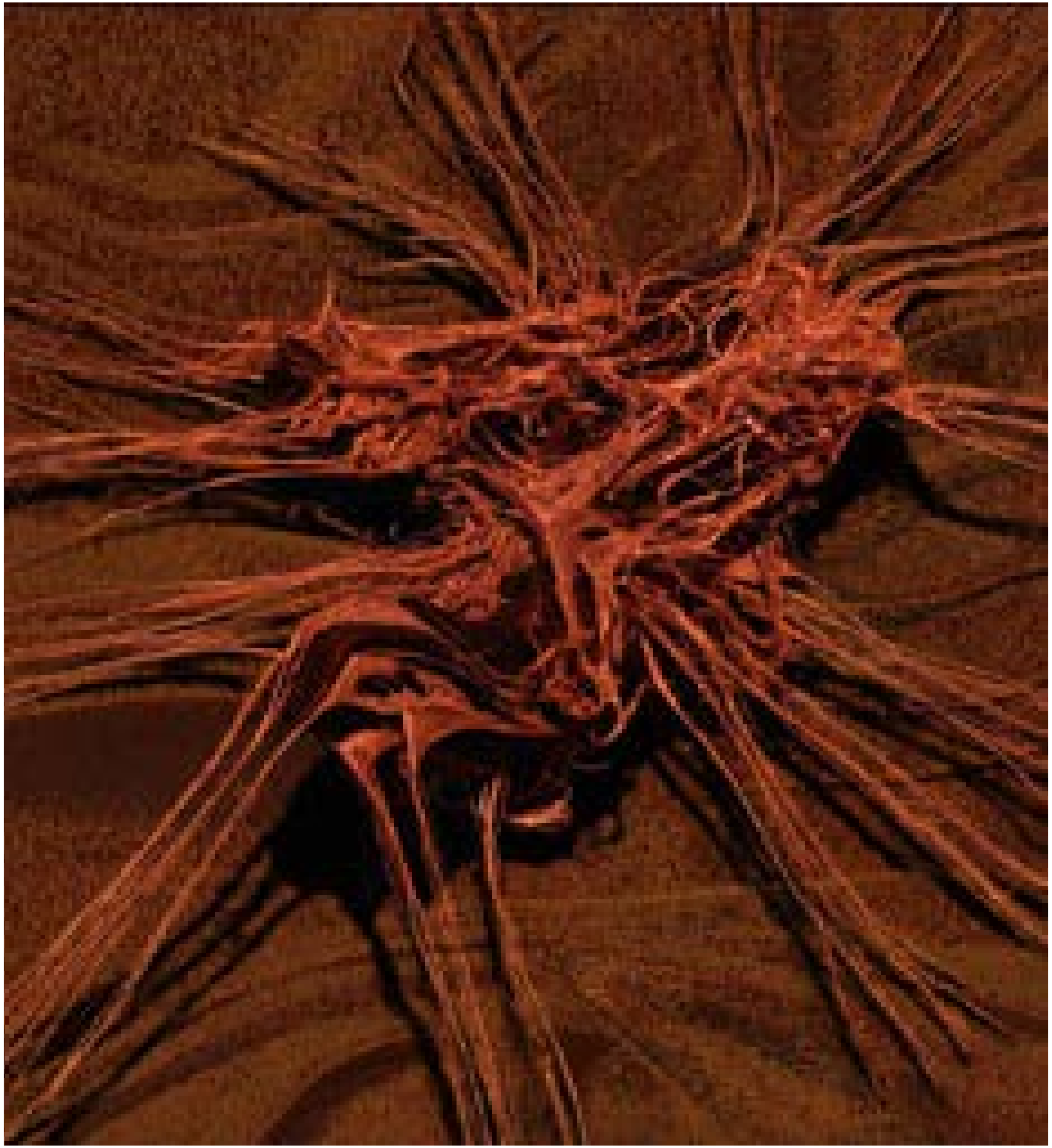
FIBROUS HOUSE

Architects: **Kokkugia**
Year: **2012**

The Fibrous House by Kokkugia is an experiment into constructing a building from a single element, the strand. The structure is inherently chaotic and embraces the complex interweaving of the strands. Swarms usually shies away from surfaces because they consists of point clouds and agent trails, The Fibrous House makes use of this logic aswell.

By having a single building component the structure foregoes the usual hierachy of elements that are usually present in buildings. The strand negotiates between structural interweaving and ornamental functions without differentiating between the two. A core theme in the project is redundancy and intensity, structural problems are taken care through sheer amount of strands and interweaving.





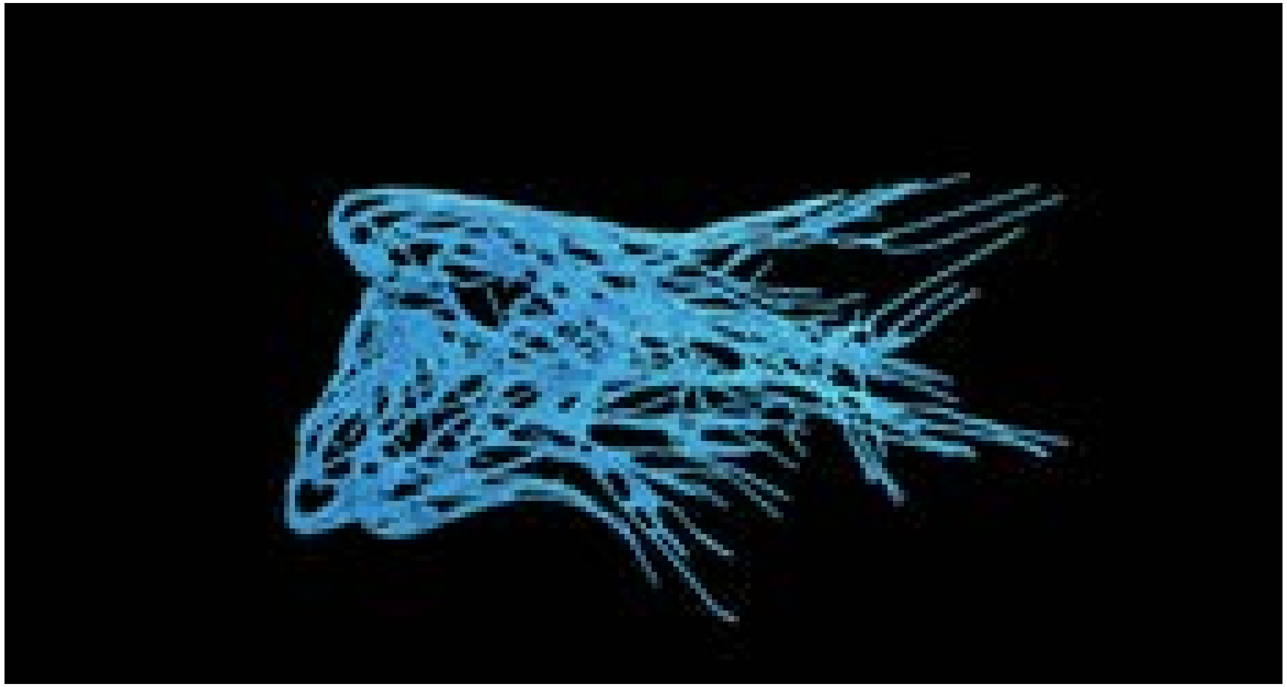
CLIFF HOUSE

Architect: **Roland Snooks**
Year: **2012**

The Cliff House in comparison with the Fibrous House has a similar underlying logic of swarms and particles but has a very different aestethic. Instead of the dizzying complexity of interweaving strands it uses surfaces with strands captured within them, this increases the legibility of the form and makes it more harmonious.

It increases the number of elements used in the construction to two but is still devoid of hiearchy as understood in the usual way. Instead hiearchy can be understood as the varying levels of intensity and density of strands and agent connections.

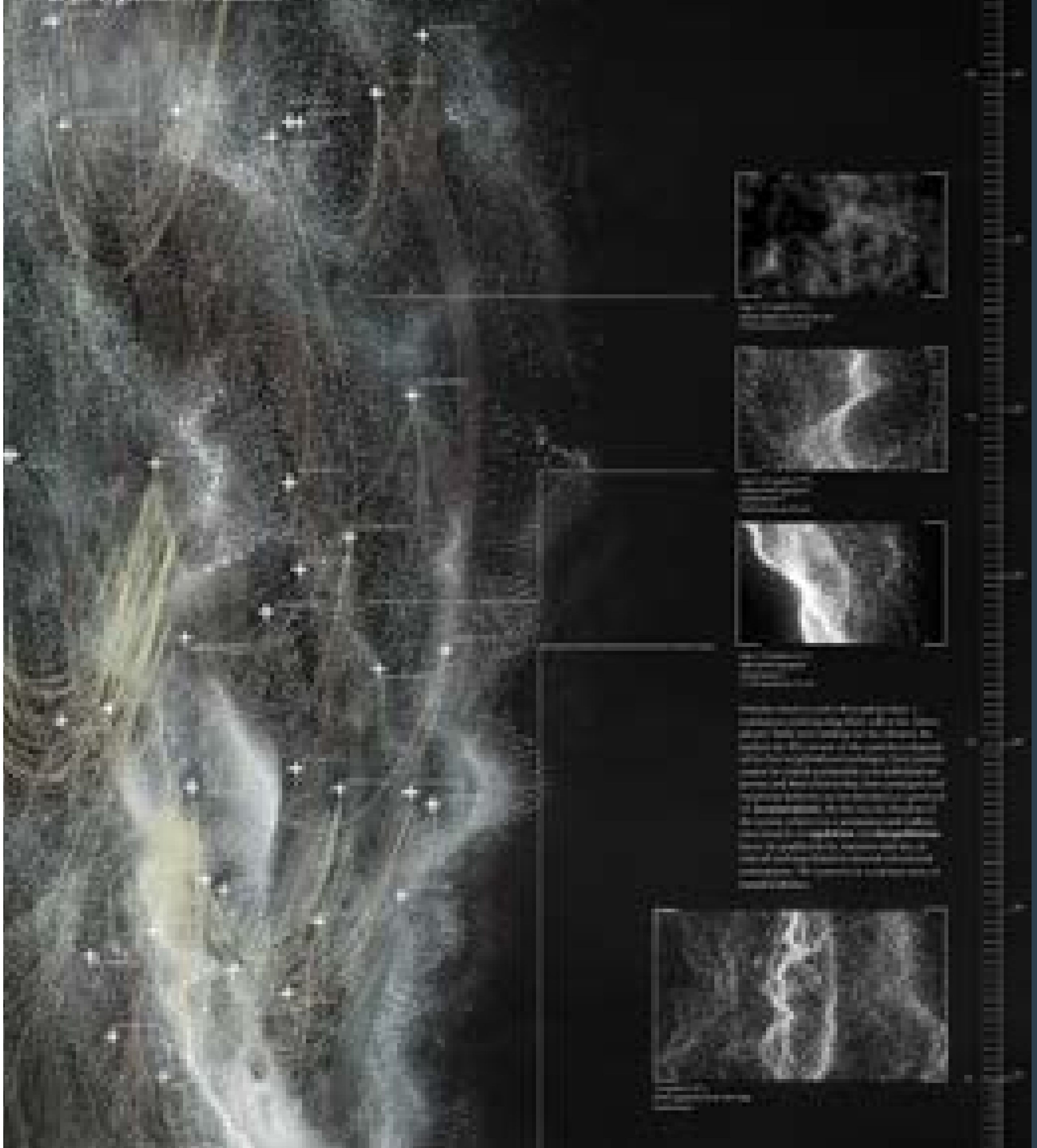
In order for a structure to be legible it cannot send out too much unstructured information, in that regard I think that this project is more successful than the previous one, the transparent surfaces increases readability and harmony.

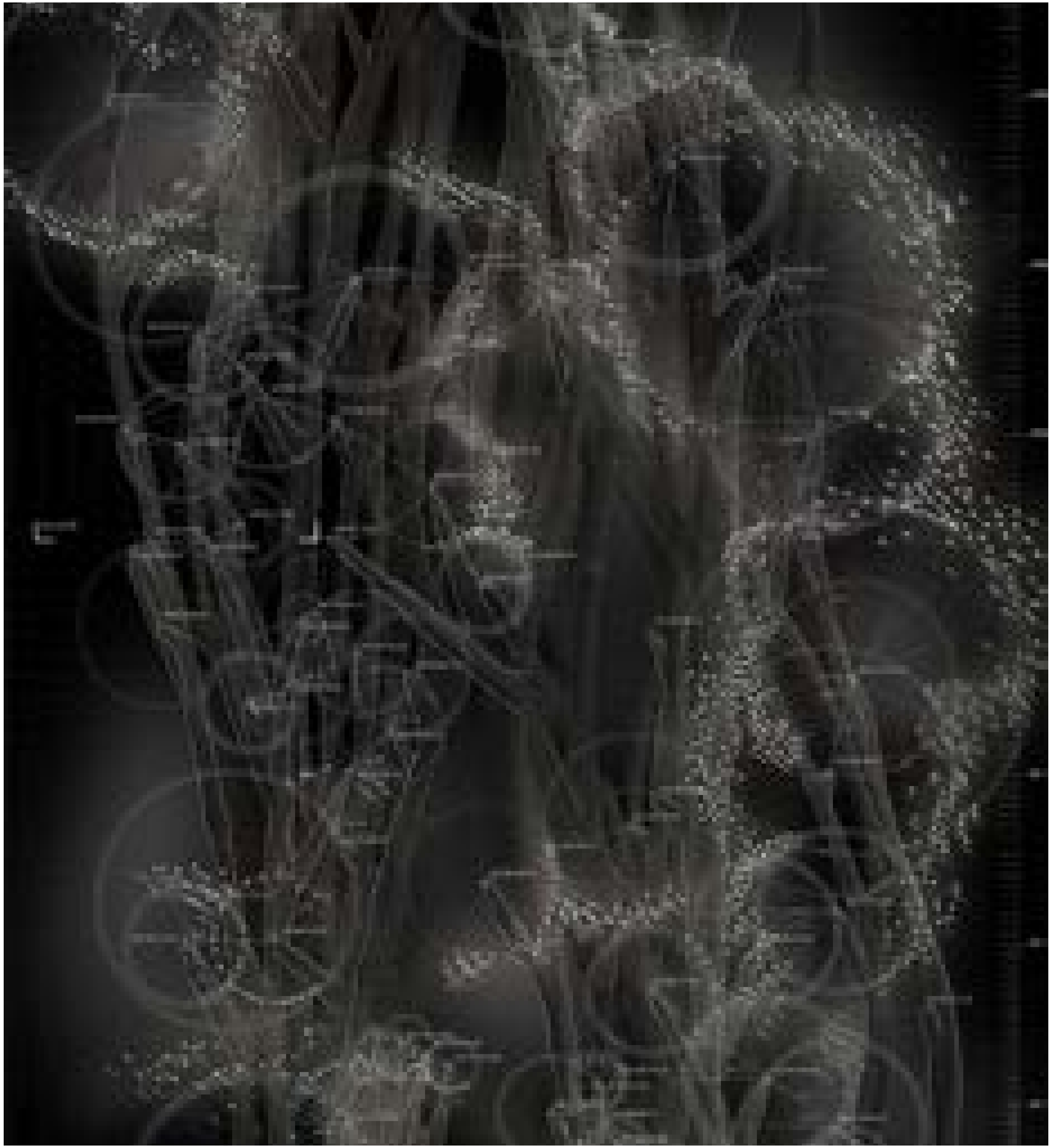


Terra(air)forma

Creators: **Evita Fanou & Nathan Hoofnagle**
Year: **2011**

Terra(air)forma is project aiming at replicating the illusory qualities of clouds by using swarm systems. Most projects that make use of swarm technology run into the problem of how to create a volume from agent movement. Fanue and Hoofnagle approaches this issue by using the agents themselves as visible particles that organize into areas with differing density and opacity. Compared to the projects by Kokkugia the aestethics foregoes the biological sensibilities and aims to replicate phenomenon such as mist and spores.





AESTHETICS OF SWARMS

Most projects that use swarm tools exhibit the visual qualitites seen in the two Kokkugia projects and the images on the opposing page. They embrace the logics of agent trails and splines to allude to biological forms. Either the trails are preserved and made thicker like in the Fibrous House or they can be made to form volumes like in the Magmatic Contingencies project, the impression is visually complex and in some ways upsetting.

I plan to take inspiration from Terra(air)forma and Cliff House by combining the varying densities of particles with how the Cliff House embeds structure within a resin-like material. A close parallell is ice, this material contains impurites and air bubbles that plays with the dynamic of opaqueness and opacity. Furthermore connections are made between different parts of the iceblock like in the top image, this effect can be replicated with connections between agents and used to denote hiearchy.

Top: Ice at Lake Uvildy - Ilya Birman

Bottom: Gas bubbles in ice at Vermillion Lake - Paul Zizka

Top Left: Magmatic Contingencies - Carson Russell & Mengna Miao
Bottom Left: DUNElab - Olga Kovrikova, Asa Darmatriaji, & Timothee Raison
Three images on the right: Excerpts from the Object-Oriented Eclecticism Workshop

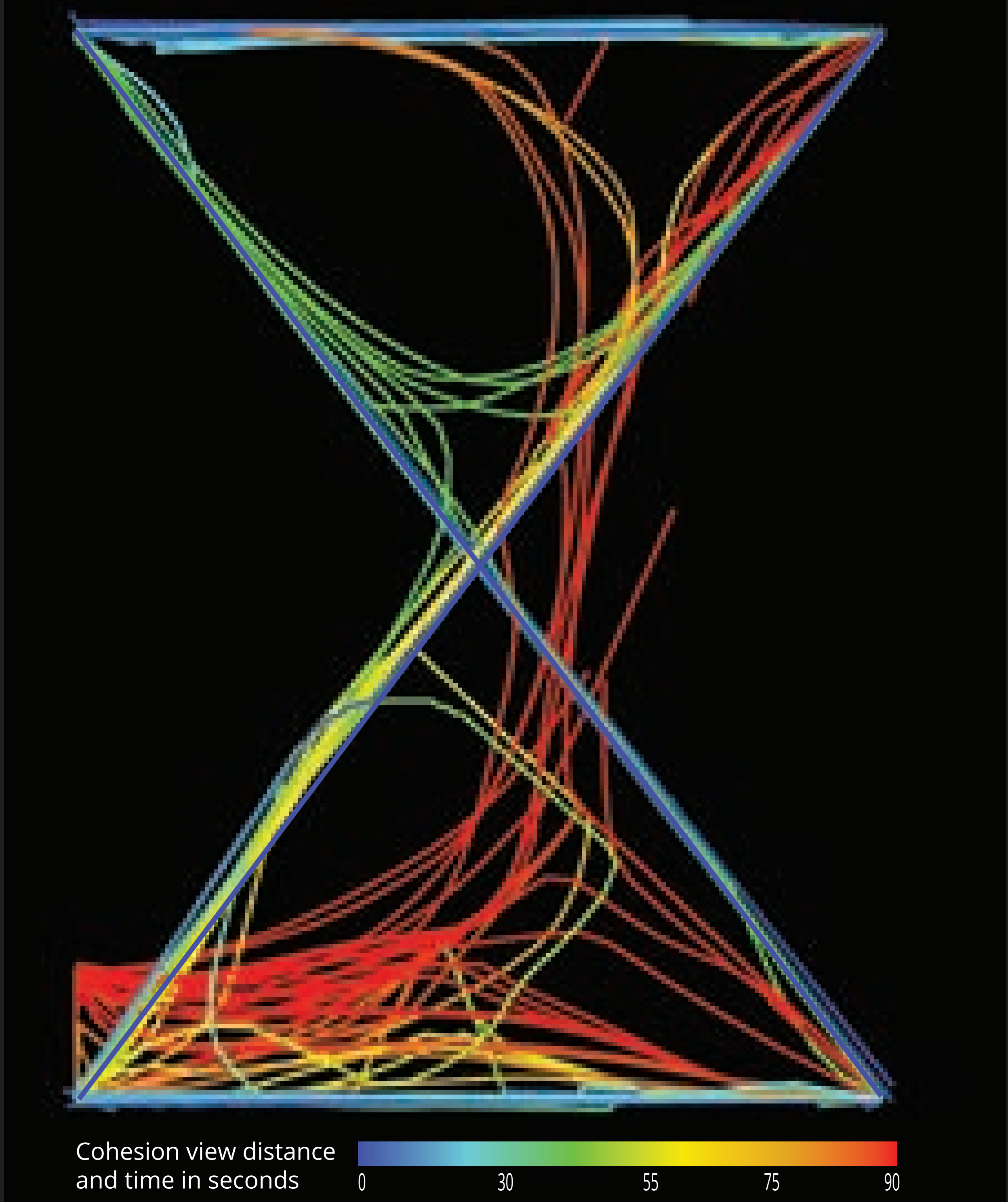
9

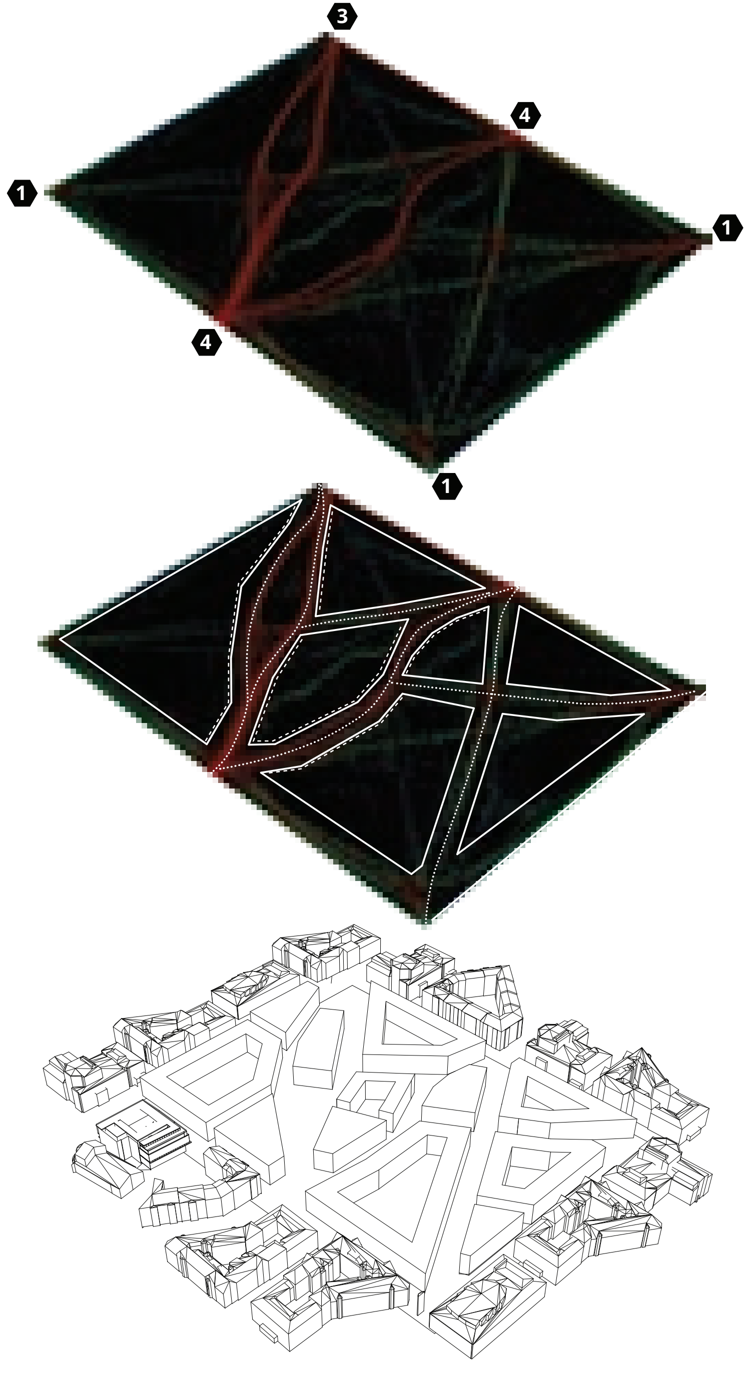
THE GENERATIVE TOOL

The behavior of simulations is dependant on the formalization of the vision and input. In this case the propensity of people (agents) to seek out other people were guided by a parameter changing the view distance of each member of the system. A value set to zero completely nullifies the stigmergic behavior while a value overpower the lure of the attractors. These high values could in some cases have agents moving in circles which over time attracted more agents, this phenomenon is also seen in ants called an “ant mill”.

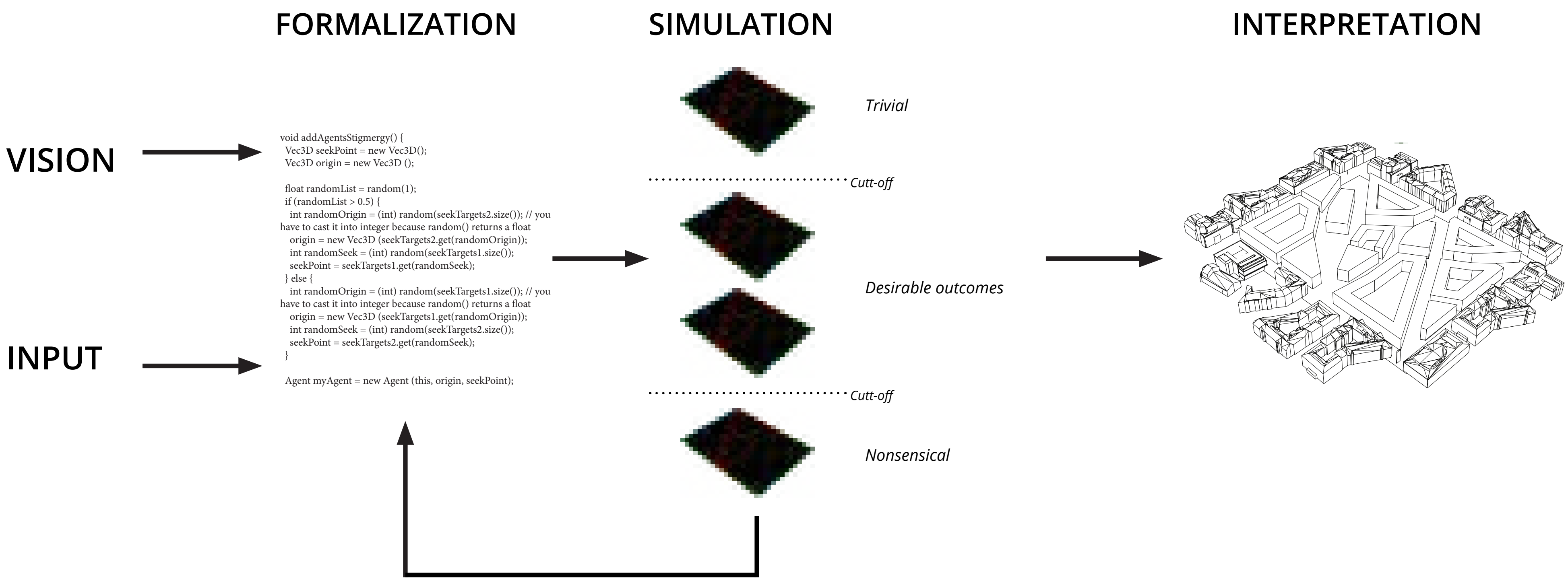
The image to the right shows the view distance parameter mapped to have the same value as the elapsed time.

Right: Aggregated points from agent movement, warmer colors indicate higher agent density





Left: Screenshot of simulation
Right: Three images showing simulation to interpretation



SIMULATION AND DESIGN INTENT

Achieving intent while using generative processes always comes back to control. The designer can influence the final result in two ways, in the formalization of the vision and input, and in the interpretation of the outcome from the simulation. The simulation results tend to fit into three categories: trivial, desirable and nonsensical. How each result is sorted into one category or the other is decided by the cutoffs. What passes for a desirable outcome is set by the design intent, in this experiment the goal was to form a network of paths that could act as the primary road network in a city environment, desirable results here meant an outcome that was plausible but not simply lines drawn between attractors. The system was continually updated both in the formalization of the design goals but also in the relative values of its parameters such as attractor strength and stigmergic properties.

In addition to the formal process of turning vision and input into code there were intuitive choices in selecting and interpreting results. In this case, zones with little or no agent movement were designated as locations to place buildings and areas which showed great intensity were marked as roads or in some cases public areas. In my view a formal approach blended with intuitive choices is the best one, and uses the strengths of computation and architectural knowledge.

Top: Volume arising from agents, same simulation over a timespan of 1 minute

INTERPRETATION

The intuitive process looks different according to the type of simulation, in the urban planning experiment the generative system were used for solving a problem with many possible solutions and negotiating between different inputs of varying strength. The simulation were used more as a suggestion or reference instead of taking the result at face value.

When generating geometry, intuitive steps can be included if the possibility of altering the product in other programs is made possible. The result or parts of it can be scaled, subdivided or altered in other ways to conform to the goals set out in the beginning. In this way a feedback process can be implemented where the simulation results is intuitively altered and inserted into the generative machine again.

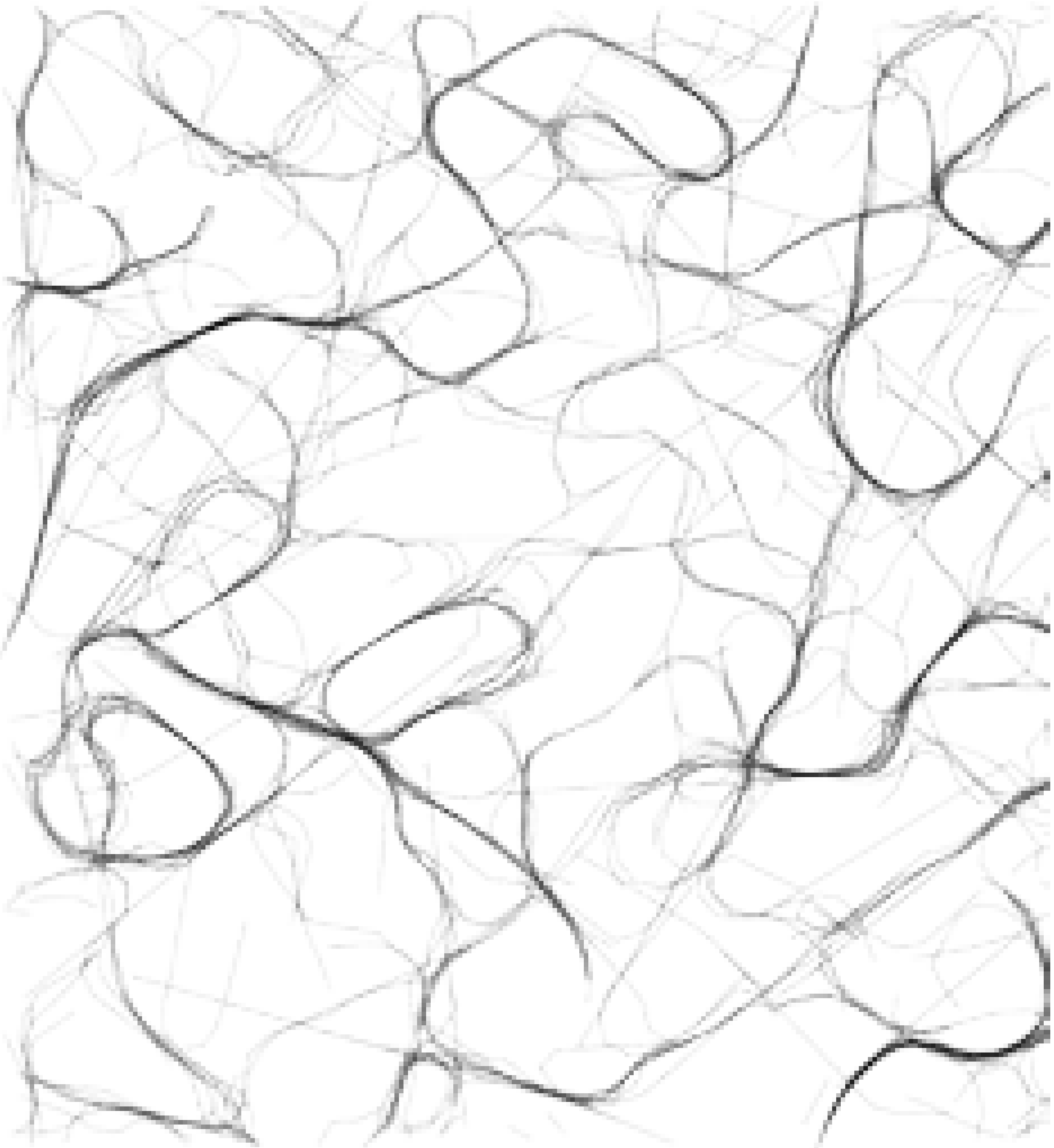
Right: Image showing swarms altering an imported doric column

10
SURFACE ARTICULATION

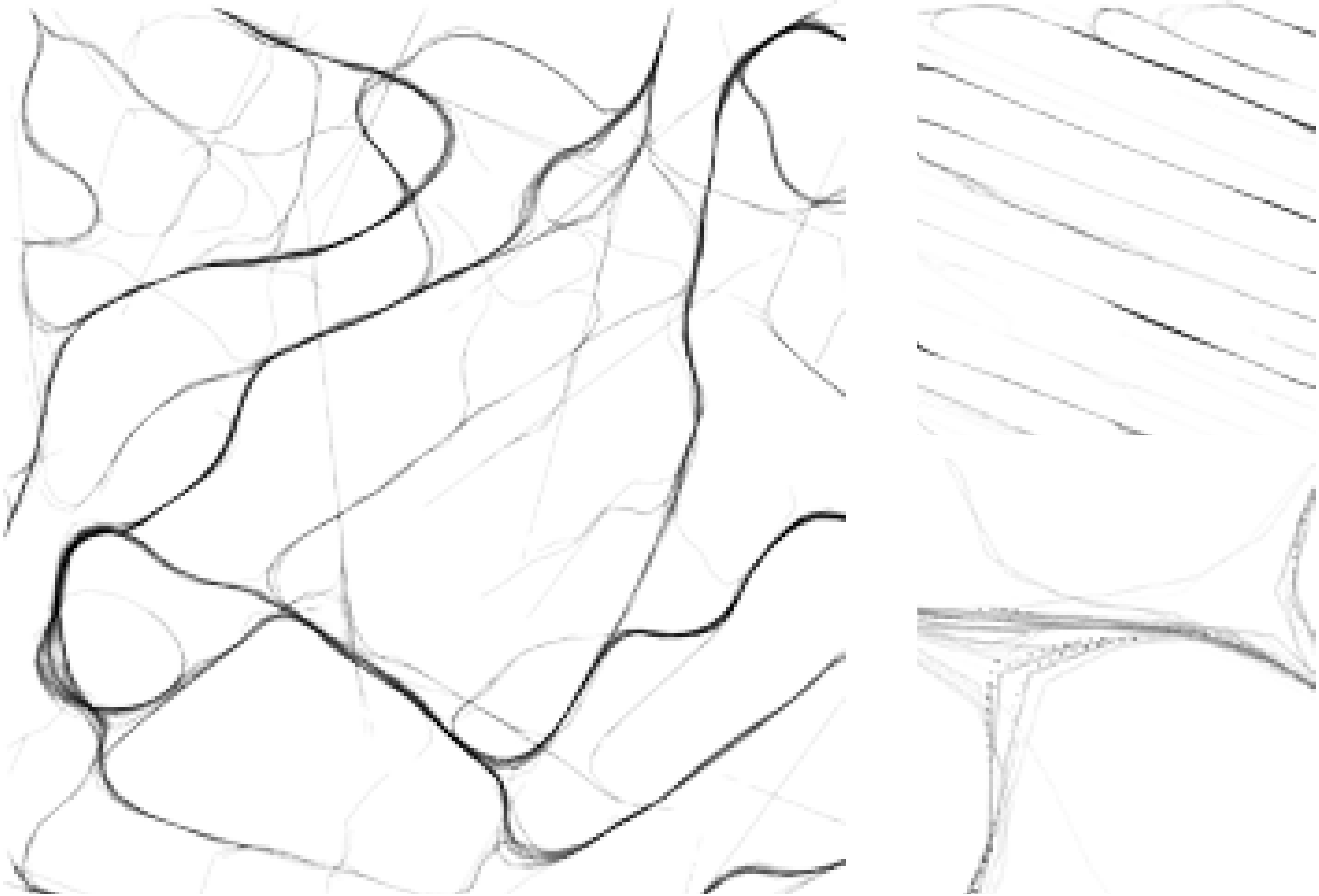
STIGMERGY

Stigmergy replicates the patterns of social insects and the use of pheromone trails. The agents leaves trails that attract other agents, over time the movement tends to solidify into patterns like the image on the right. The pattern is unique for each simulation and changes over time.

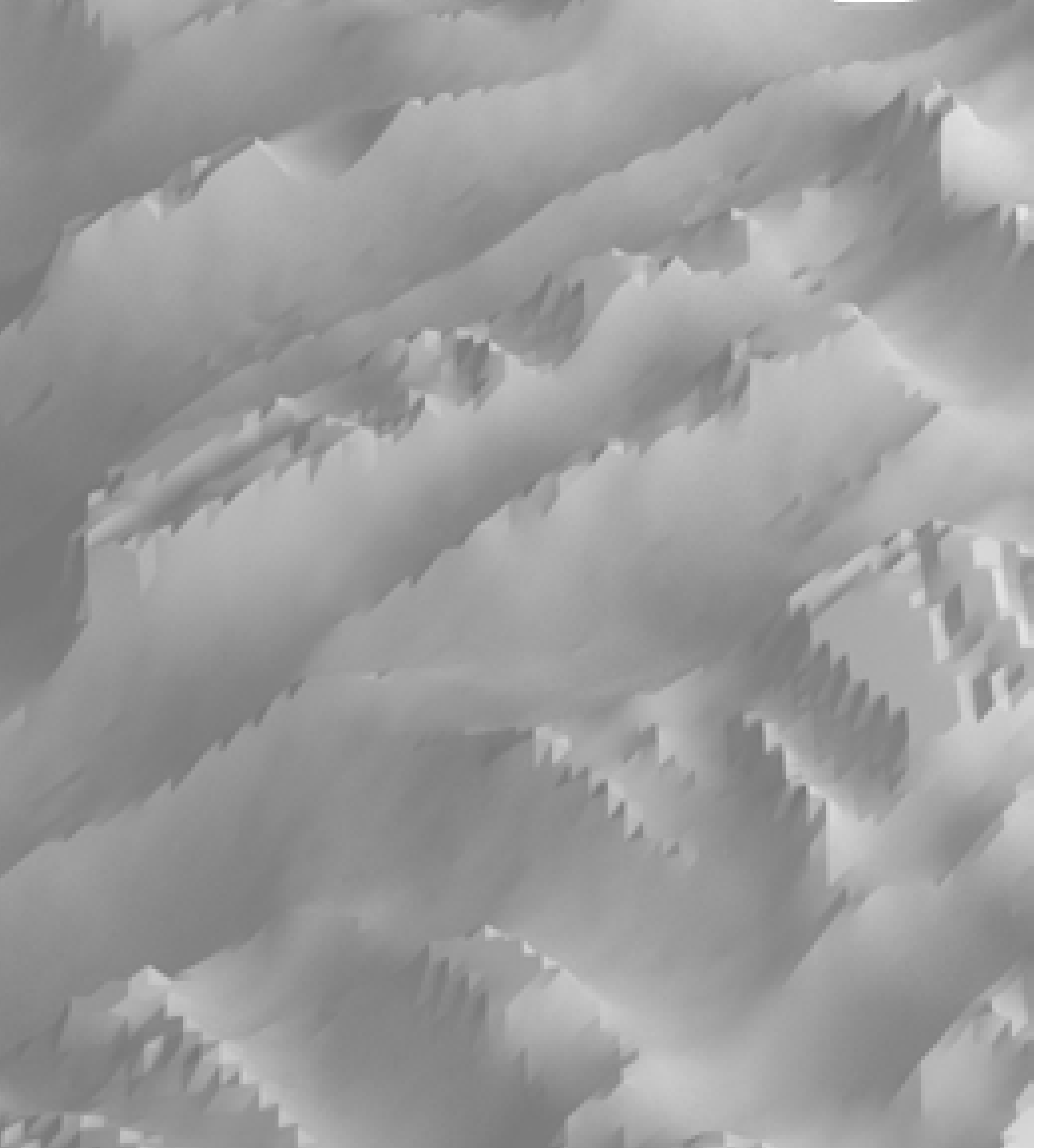
Patterns are easier to understand when they can be tied to an underlying logical system like stigmergy, especially still images like this outside of simulations. One can start to make sense of the underlying principles by just looking at an image of the phenomenon.



Right: Simulation, 5 seconds elapsed



Left: Simulation, 10 seconds elapsed
Top Right: Simulation, 575 seconds elapsed
Bottom Right: Zoom of simulation, 45 seconds elapsed

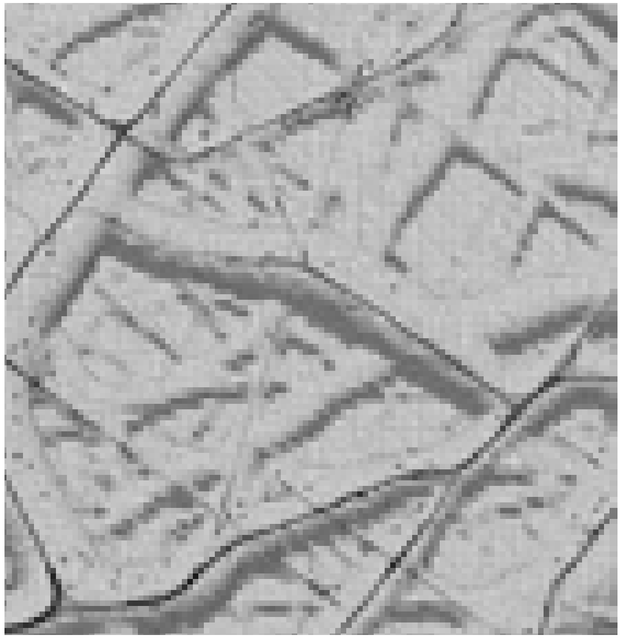


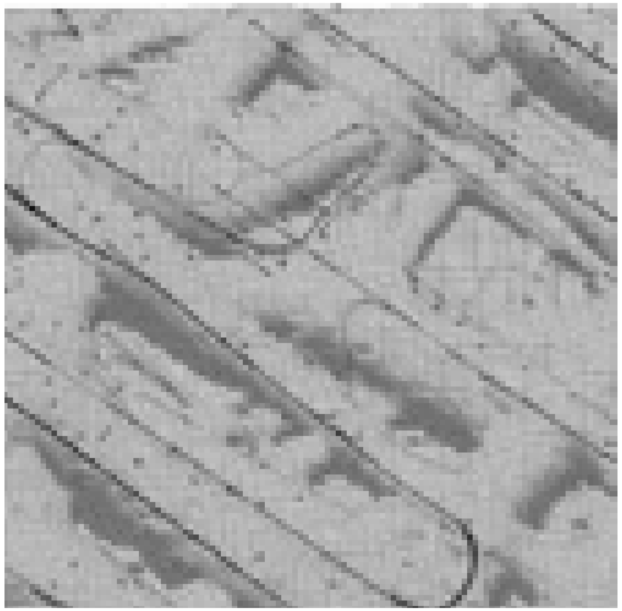
AGENT EROSION

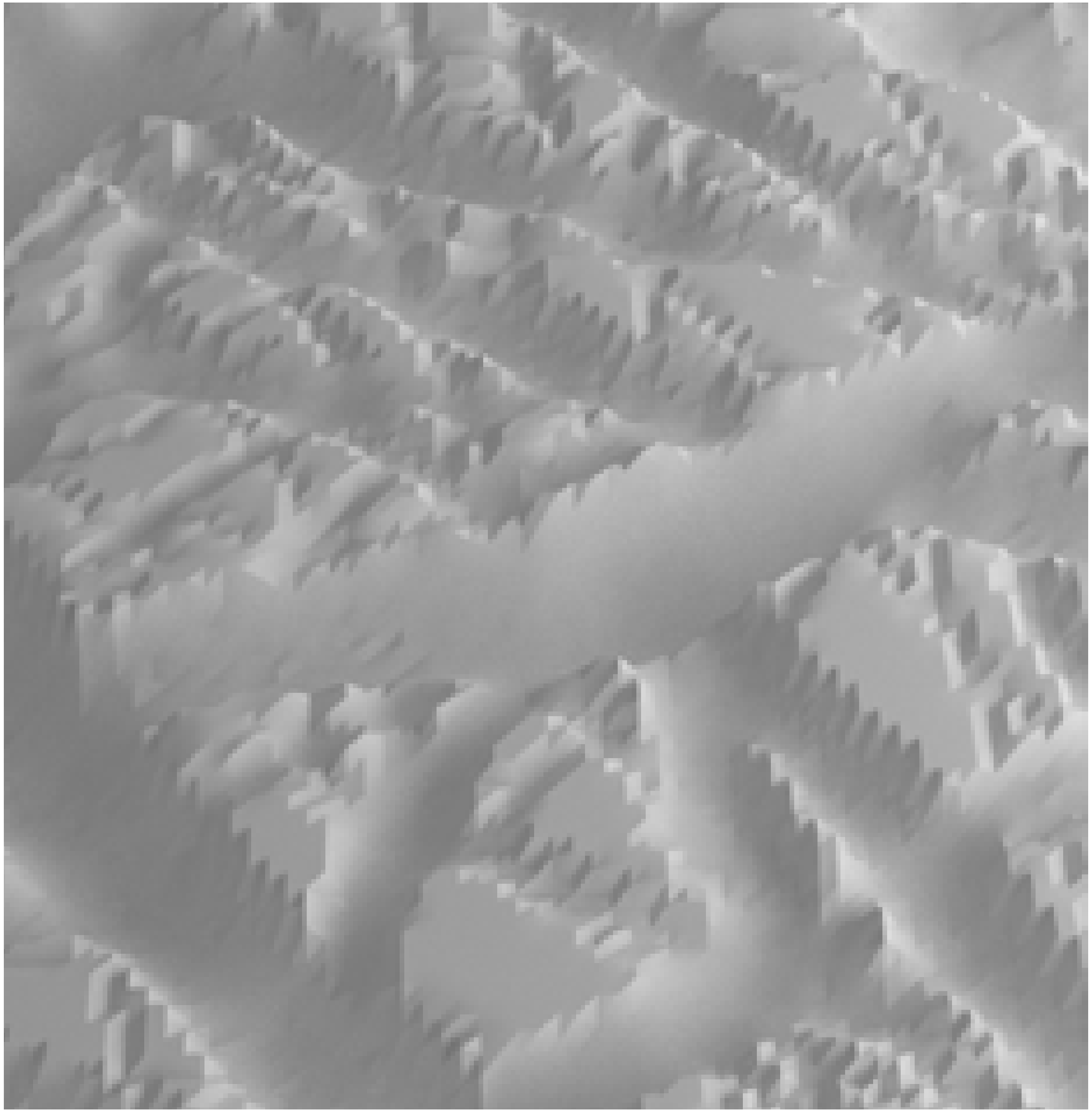
After implementing the stigmergic behavior in the agents the next step was to make them sculpt a surface. The stigmergic patterns are be made visible in three dimensions instead of just existing in two dimensions.

This function lowers the z-coordinates for the underlying grid points based on agent movement. Each time an agent moves it lowers the nearest point in the grid by a certain amount. Because the agent selects the nearest point and not the one beneath it, smooth valleys are formed in the landscape that contrasts against the more jagged parts. The resulting surface is markedly influenced by the stigmergy pattern.

Left: Zoom in of landscape, 249 seconds elapsed







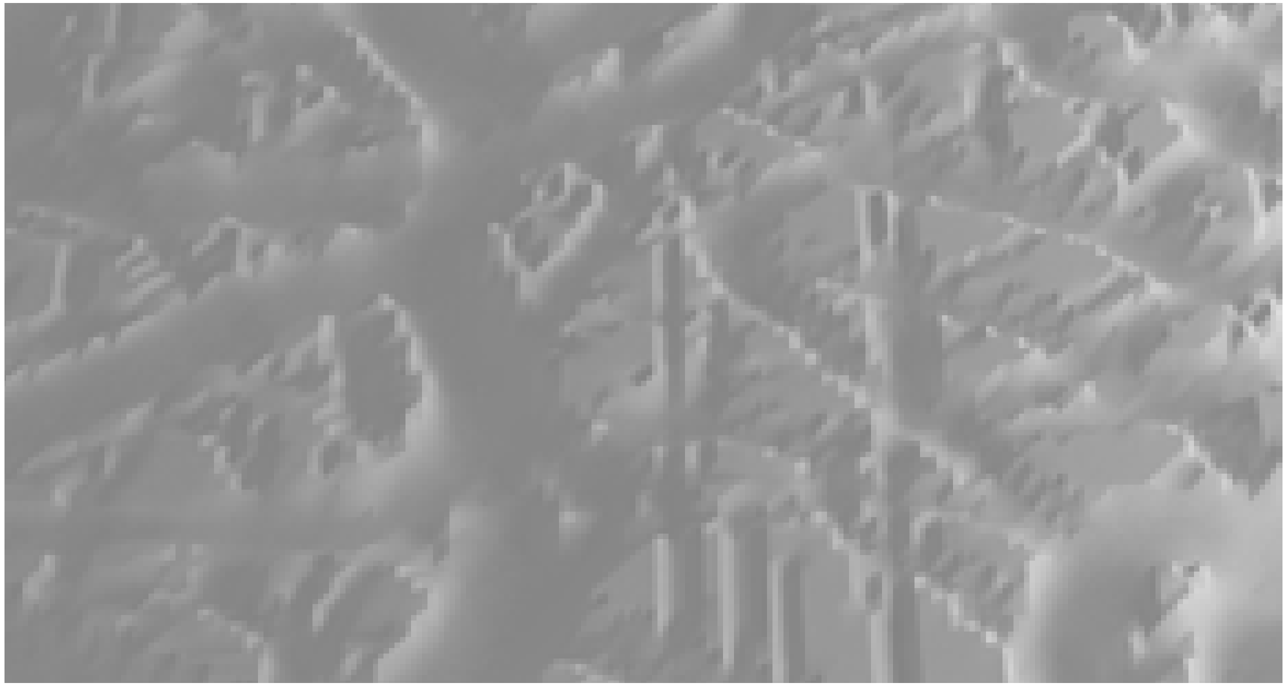
Top Left: Simulation, 111 seconds elapsed
Bottom Left: Simulation, 249 seconds elapsed
Right: Render of landscape, 111 seconds elapsed

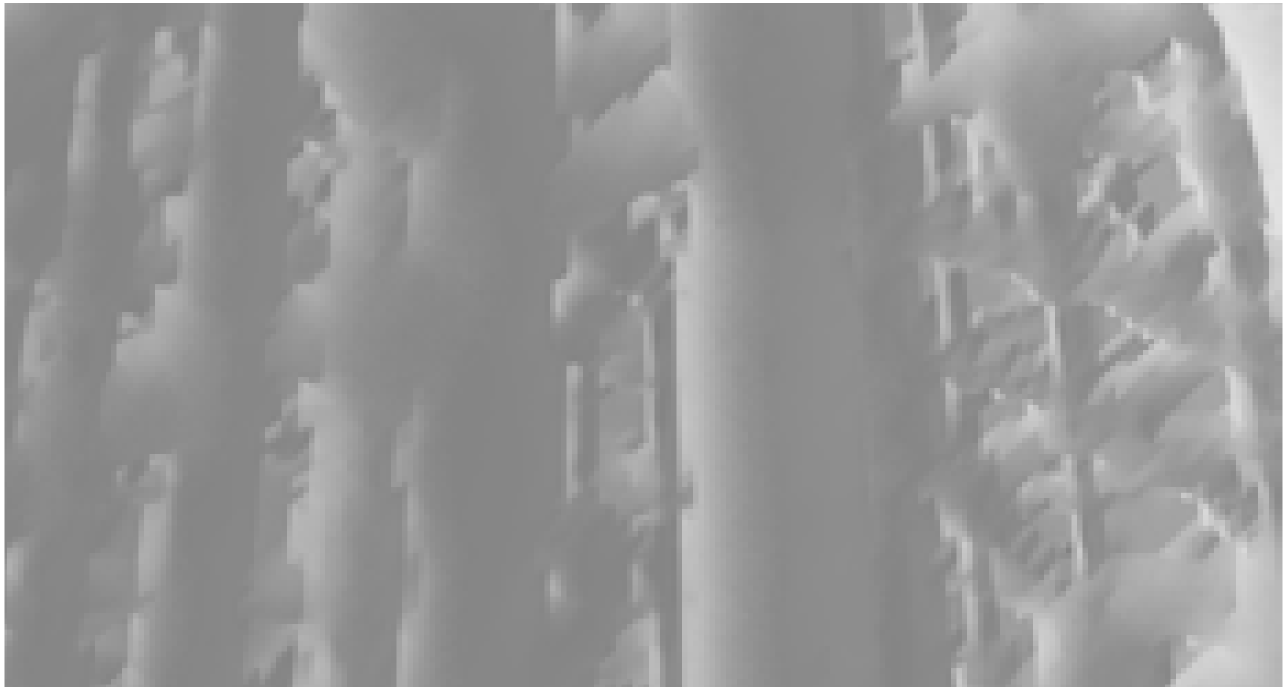
SIDEWAYS GRAVITY & SWARM CONCENTRATIONS

The Sideways Gravity function applies a downward force to the movement of the agents which is controlled by a slider inside the simulation. The effect gets more pronounced as time goes by which can be seen in the two images. In order to get interesting results one needs to increase the force after the first behavior has been active for awhile. To make the most of this code I propose to add a function that changes the separation value of the agents according to their y-position.

The Swarm Concentrations script identifies local concentrations of agents and forms centers that can be seen as large red dots in the simulation images. These centers attract nearby points in grid and adds a random value to their z-coordinate. It also freezes their position to prevent the agent erosion from erasing the change to the surface.

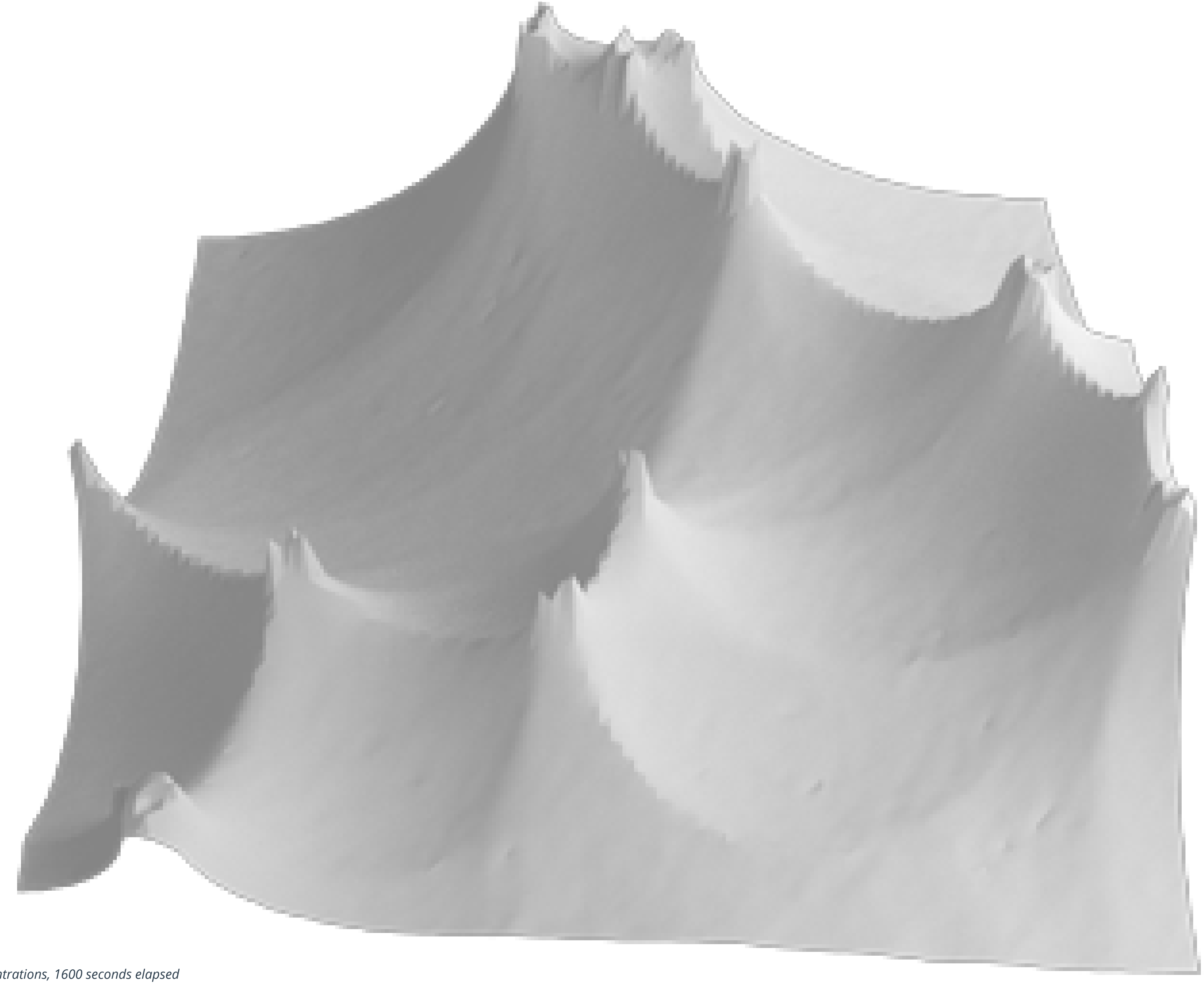
What's interesting about the two functions working together is that they form jagged and smooth parts together and also introduces more variations in height.





Top: Downward Force, 71 seconds elapsed

Bottom: Downward force, 158 seconds elapsed



Local concentrations, 1600 seconds elapsed

CODE EXCERPT

This is one part of the Agent Concentrations behavior. This script recieves the location of a cluster of agents and moves the points in the surface closer to it. The original idea behind this function was to change the surface points not just in depth but their x & y coordinate as well. What was more important was the addition of **freezing** the points once that they had moved. This produced variations in the surface expression to include jagged peaks and smooth valleys.

How this script behaves can be adjusted by changing the *"distancePointMove"* variable to control how far it will search for points close to the agent concentration. The Agent Erosion script competes against this one for influence over the surface. A high number in "*distancePointMove*" would make the script more dominant.

Furthermore, the actual displacement of the points can be adjusted by tinkering with *"pointMoveMagnitude"* and the change in depth.

```
void movePointsCenter(Vec3D _sum) {

 Vec3D sum = new Vec3D(_sum);
 for (int i = 1; i<cols-1; i++) {            //-1 to keep frame intact
  for (int j = 1; j<rows-1; j++) {

   float distance = allPts[i][j].loc.distanceTo(sum);
   float move = distance / 2;

   if (distance < distancePointMove && allPts[i][j].hasMoved == false) {

    Vec3D direction = new Vec3D(sum.sub(allPts[i][j].loc));
    direction.normalize();
    move = move * pointMoveMagnitude;
    direction.scale(move);
    allPts[i][j].loc.addSelf(direction);
    allPts[i][j].loc.z = allPts[i][j].loc.z + random(0,10);

    allPts[i][j].hasMoved = true;  //each point can only be moved once

    strokeWeight(10);
    point(allPts[i][j].loc.x, allPts[i][j].loc.y);
   }
  }
 }
}
```

recieves the agent concentration point

goes through each point in the grid

checks if point is close enough to be moved

checks if point has been moved before

moves point closer to center according to distance to center

randomizes the z-coordinate of the point to get a surface effect

the point can now not be moved again by any script

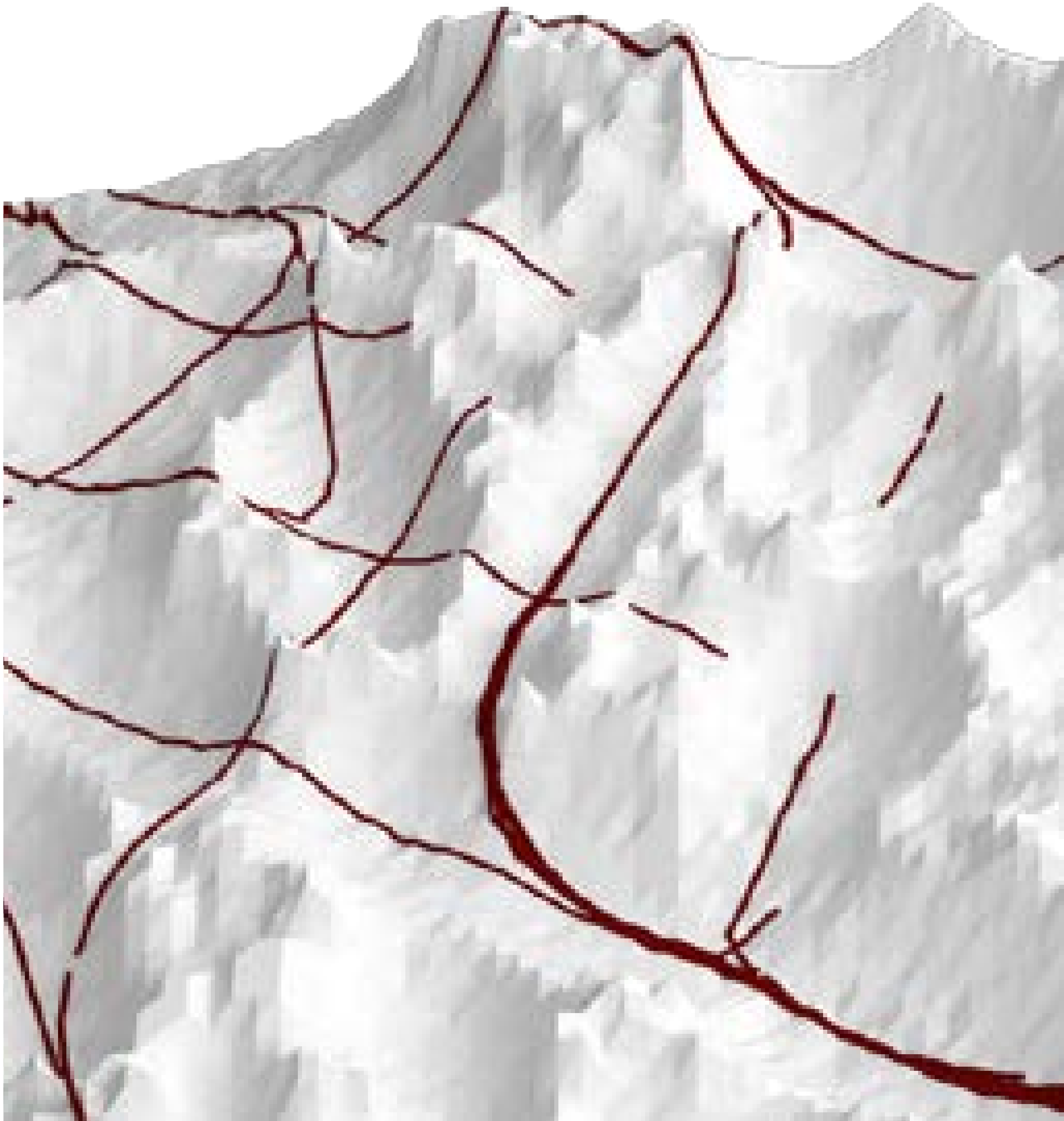
display a red circle at the new location of the point to elucidate what's going on

ADDING STIGMERGY TO MESH

To visualize the agent movement in addition to their erosion of the mesh the stigmergic patterns were given physical form. This was done in Rhinoceros by piping the curves and projecting them onto the mesh. Concentrations of lines and the depth of the surface tells a story of how the agents interact with each other and the surface.

This is a way to communicate with the users through the surface and to tell something of the underlying process behind its creation.

Right: Render of mesh, 311 seconds elapsed

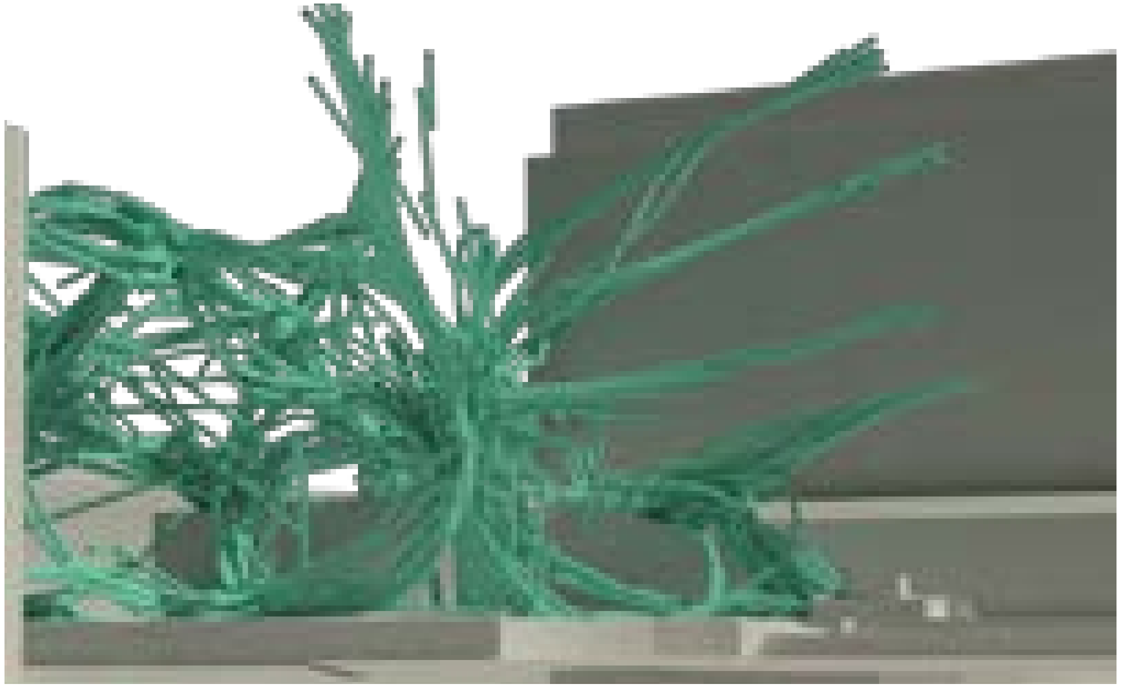


11
VOLUMES FROM SWARMS

AGENT PATHS MADE VISIBLE

From working with surface articulation I moved to three dimensional space and reconfigured the stigmergic behavior to work in 3D. The logic of swarm systems give you point clouds and curves resulting from agent movement. In this example I used the agent trails which I piped to give form to their movement.

The pipes alludes to the organic forms of sinews and roots of plants. By using points that attract and repells agents architectural qualities could be created such as inside and outside. Much like the two Kokkugia projects there is also a different understanding of what hiearchy means, the form is ordered by varying levels of intensity and movement.



Top: Downward Force, 71 seconds elapsed

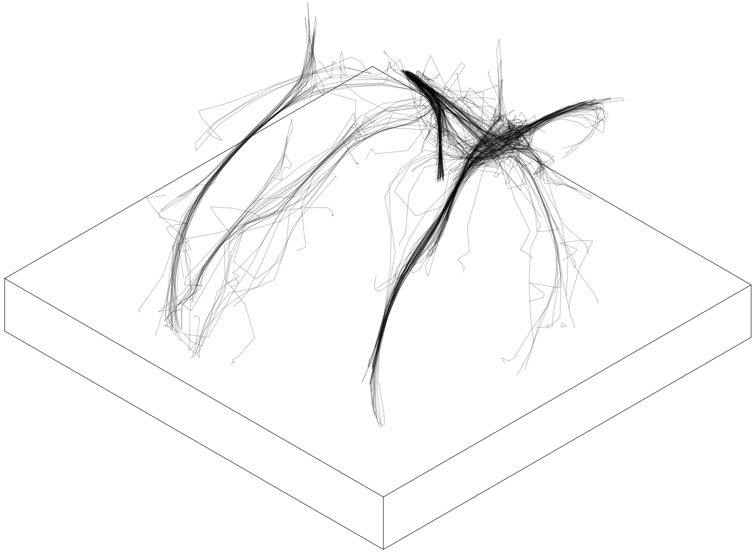
Bottom: Downward force, 158 seconds elapsed

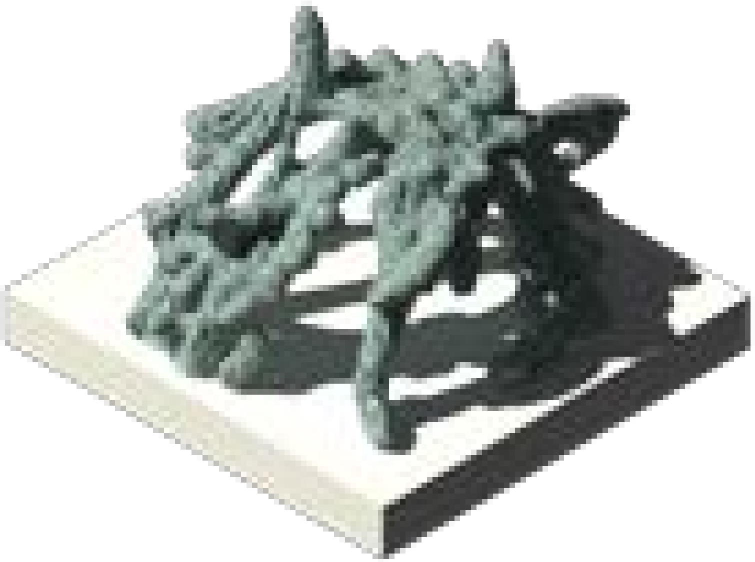
VOLUME THROUGH COCOON

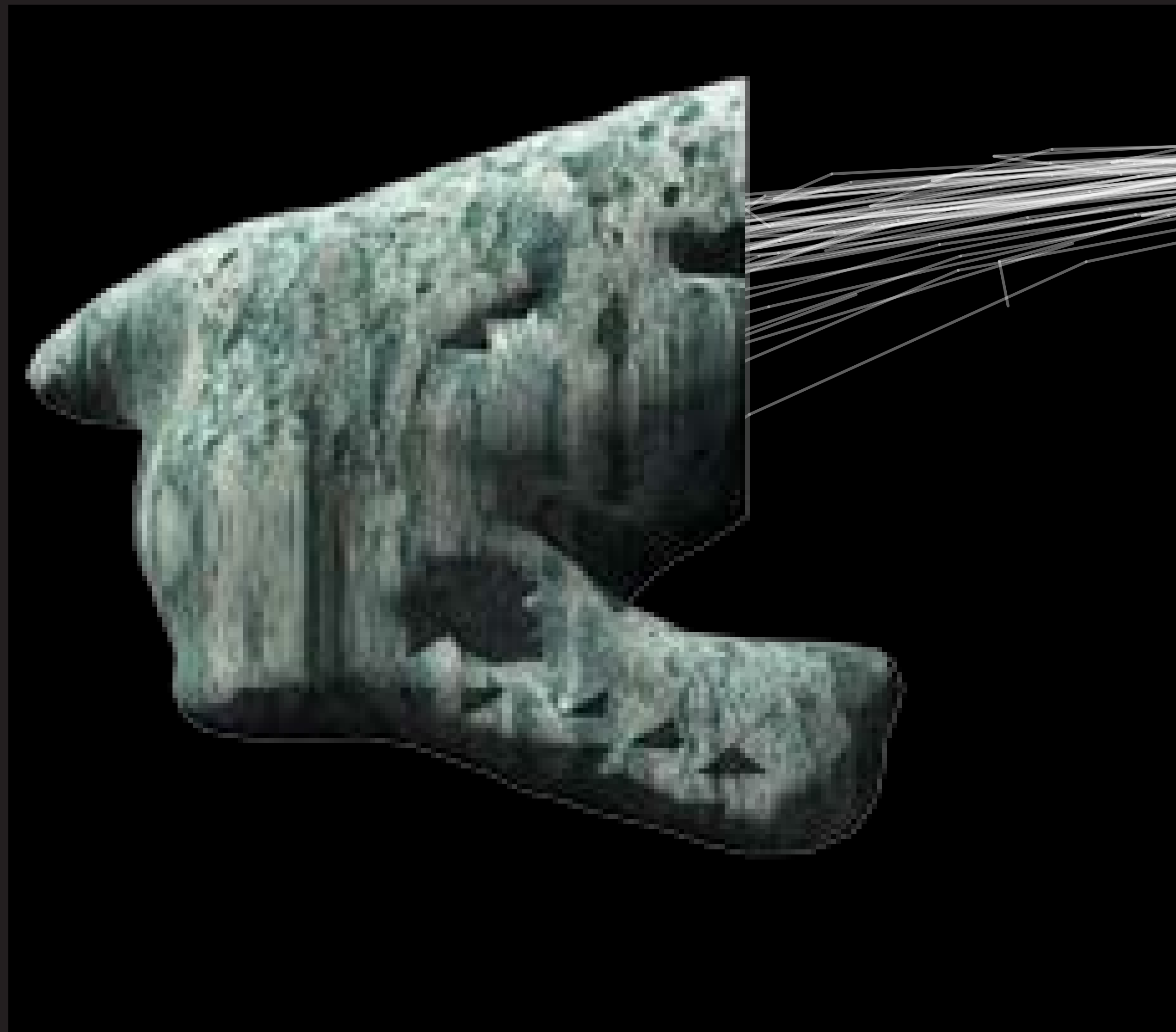
In this experiment I sought other ways to make volumes from the scripts than simply piping the paths. I used a grasshopper plugin called Cocoon which wraps geometry with a mesh. The software is quite unweildy and is very slow in producing a result. It is frustrating that computer power can be a limiting factor and to tackle this problem I reduced the amount of information in my simulation by having fewer agents moving around.

Compared to piping the agent paths this method reduced the amount of information and increased legibility by using surfaces instead of curves. One can compare this difference to Fibrous House by Kokkugia and Cliff House by Roland Snooks.

Left: Render of agent paths run through Cocoon
Opposite: Volumes in a site context







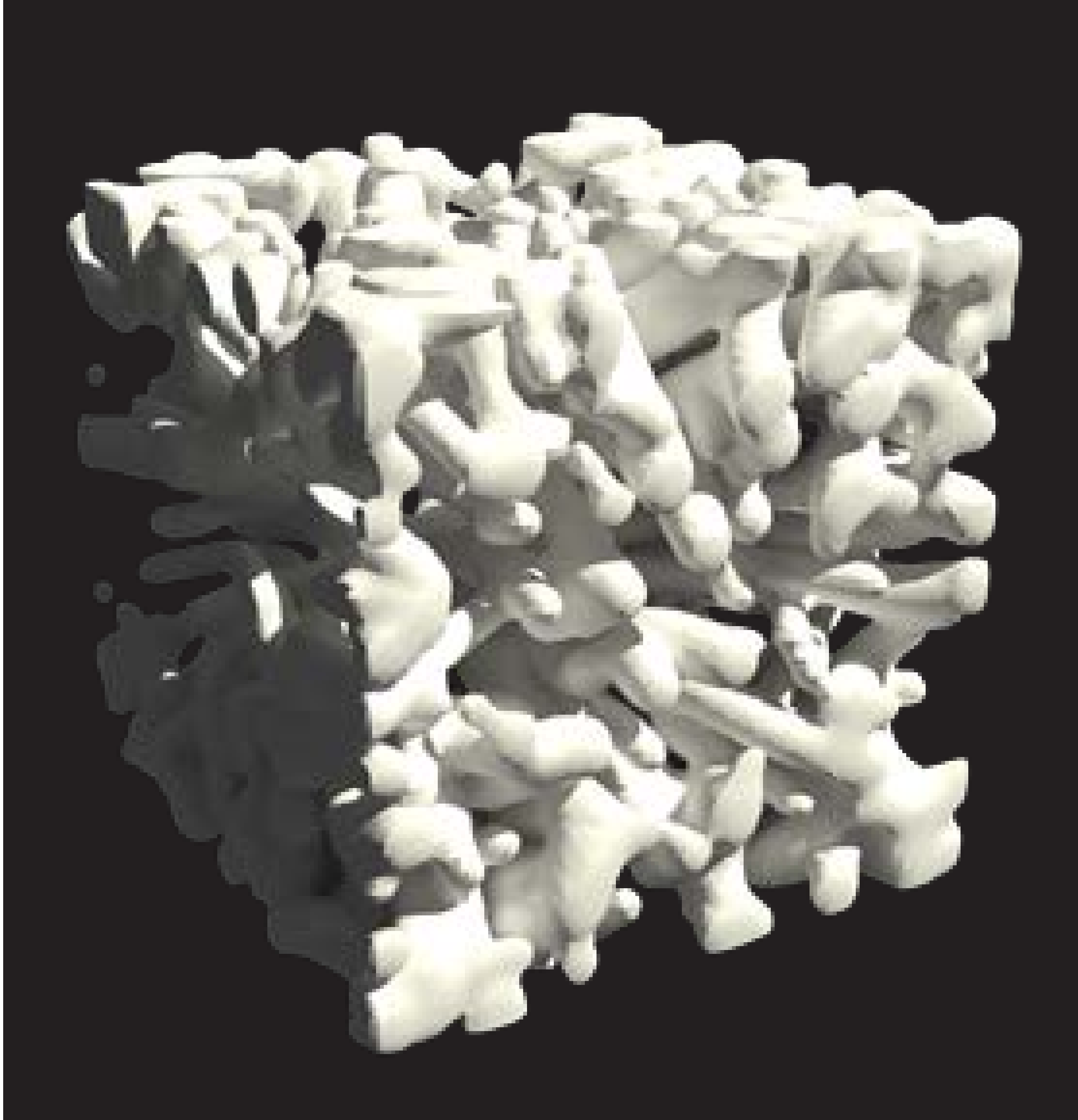
Top Left: Stigmergic paths
Bottom Left: Paths transformed into a volume
Right: Drawing showing linework and volume

12
VOLUMES IN PROCESSING

I moved on from Cocoon to try to create volumes directly in Processing. From the first moments an incredible increase in speed was visible, this made it possible to easily try out different shapes and get immediete feedback. It also shifted from static structures to moving simulations.

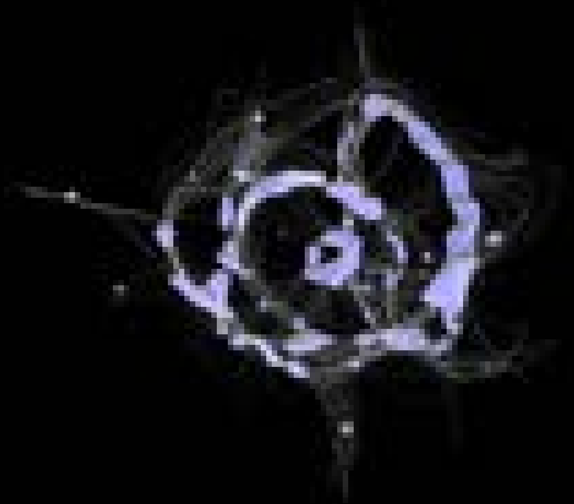
Another great addition was the possibility of having much more agents moving about and not having to worry about computing time. In processing I borrowed an algorithm called "marching cubes" that identifies clusters of points (agent locations in my case) and forms volumes around them. A problem is that the process is quite mysterious and difficult to control. I added a user interface to adjust brushsize which sets the threshhold for when volumes are formed, in addition a slider for iso was added that controlled the thickness of the volumes.

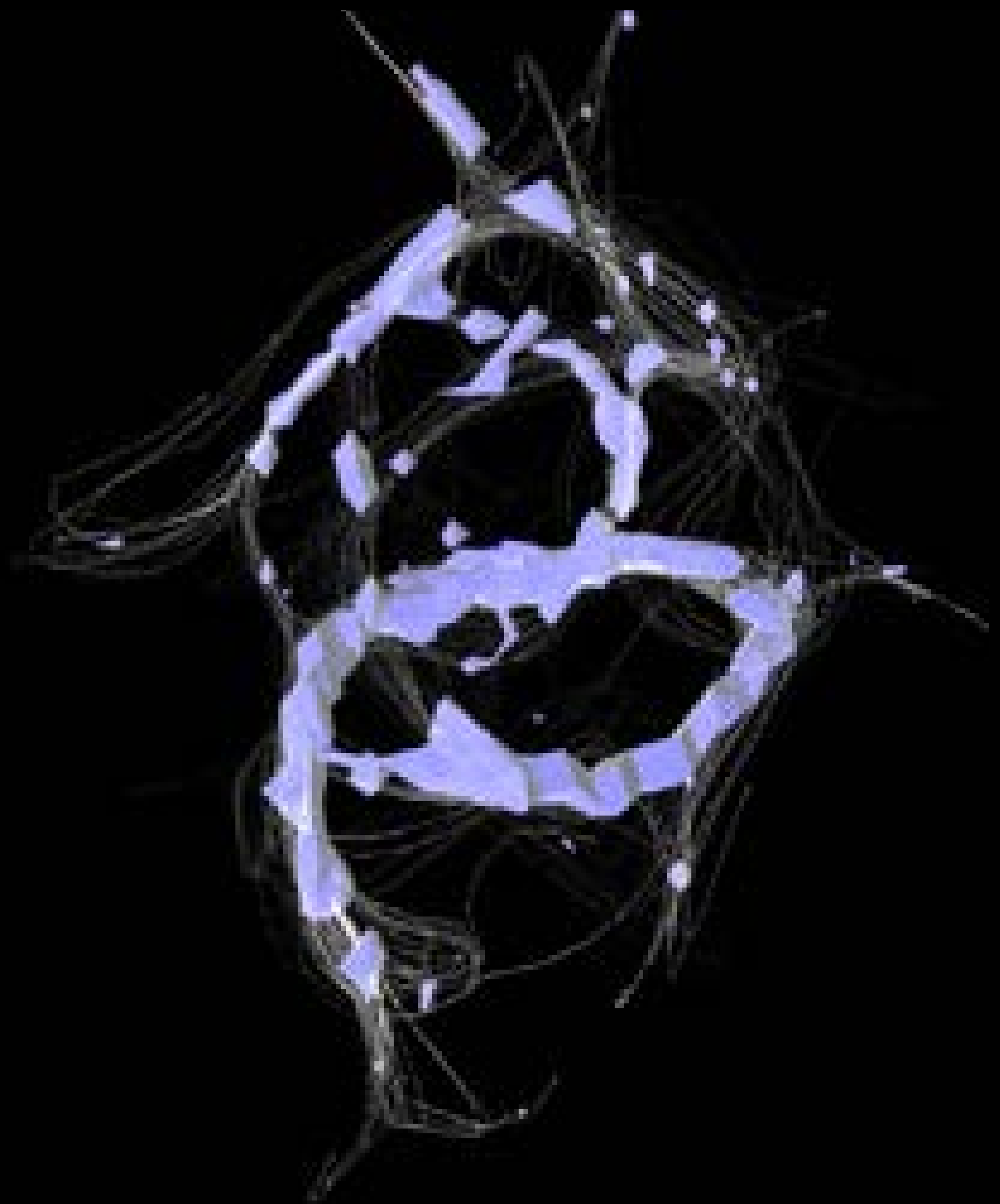
Top: Volume arising from agents, same simulation over a timespan of 1 minute

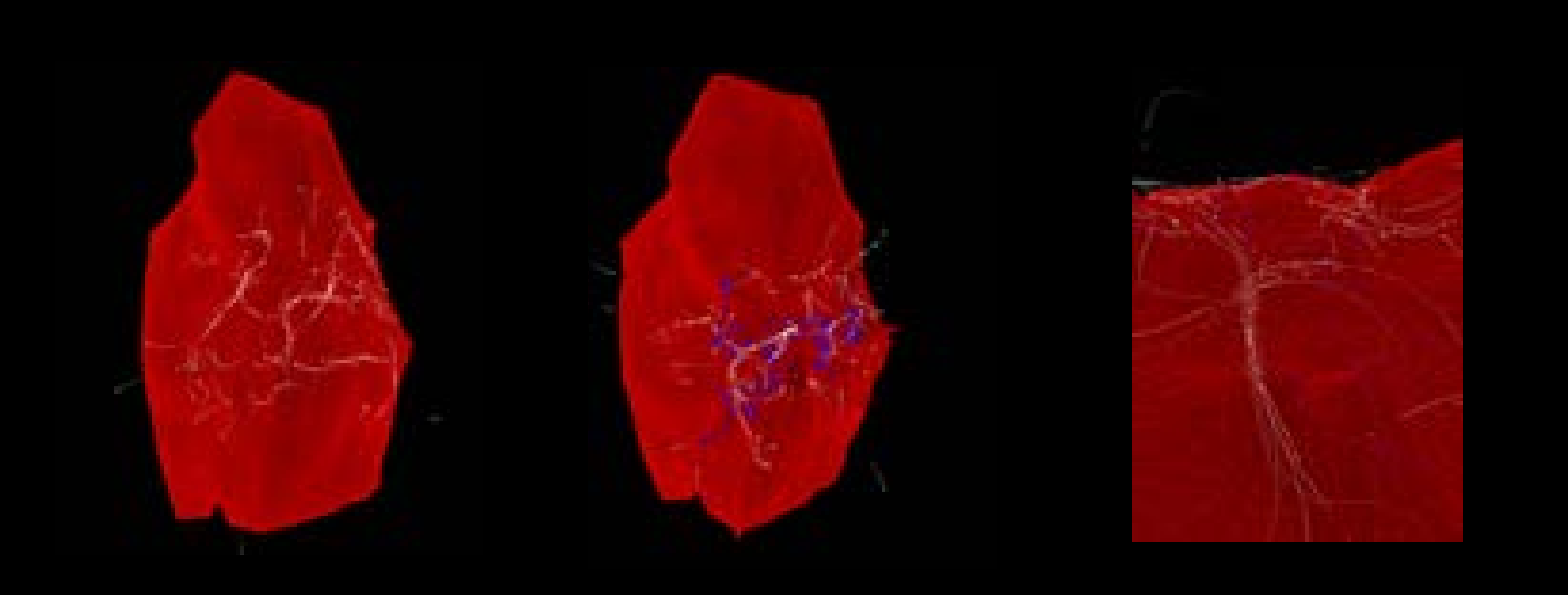


Top Left: low brushsize, high iso, 94 seconds elapsed
Bottom Left: medium brushsize, high iso, 95 seconds elapsed
Right: high brushsize, low iso, 94 seconds elapsed

The forms on this spread are created by having a low brush size and a low iso. This manages a negotiation between curves and matter and avoids the problem of too much information.



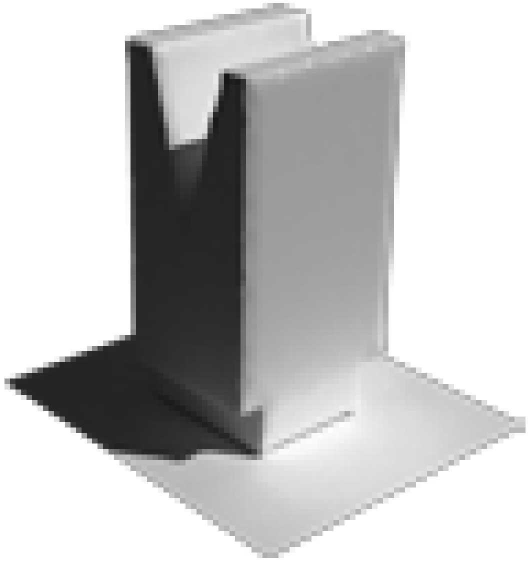


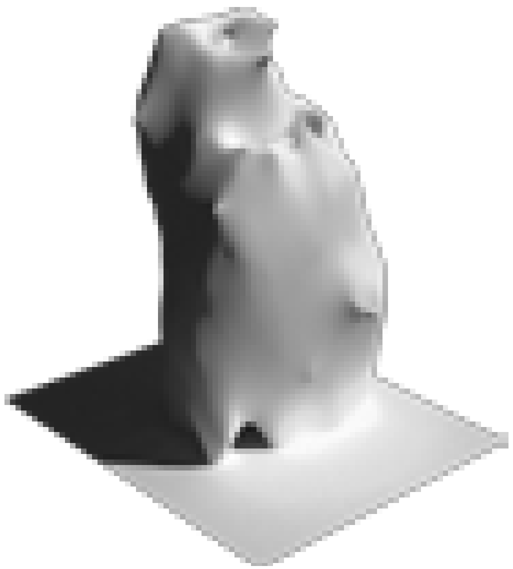


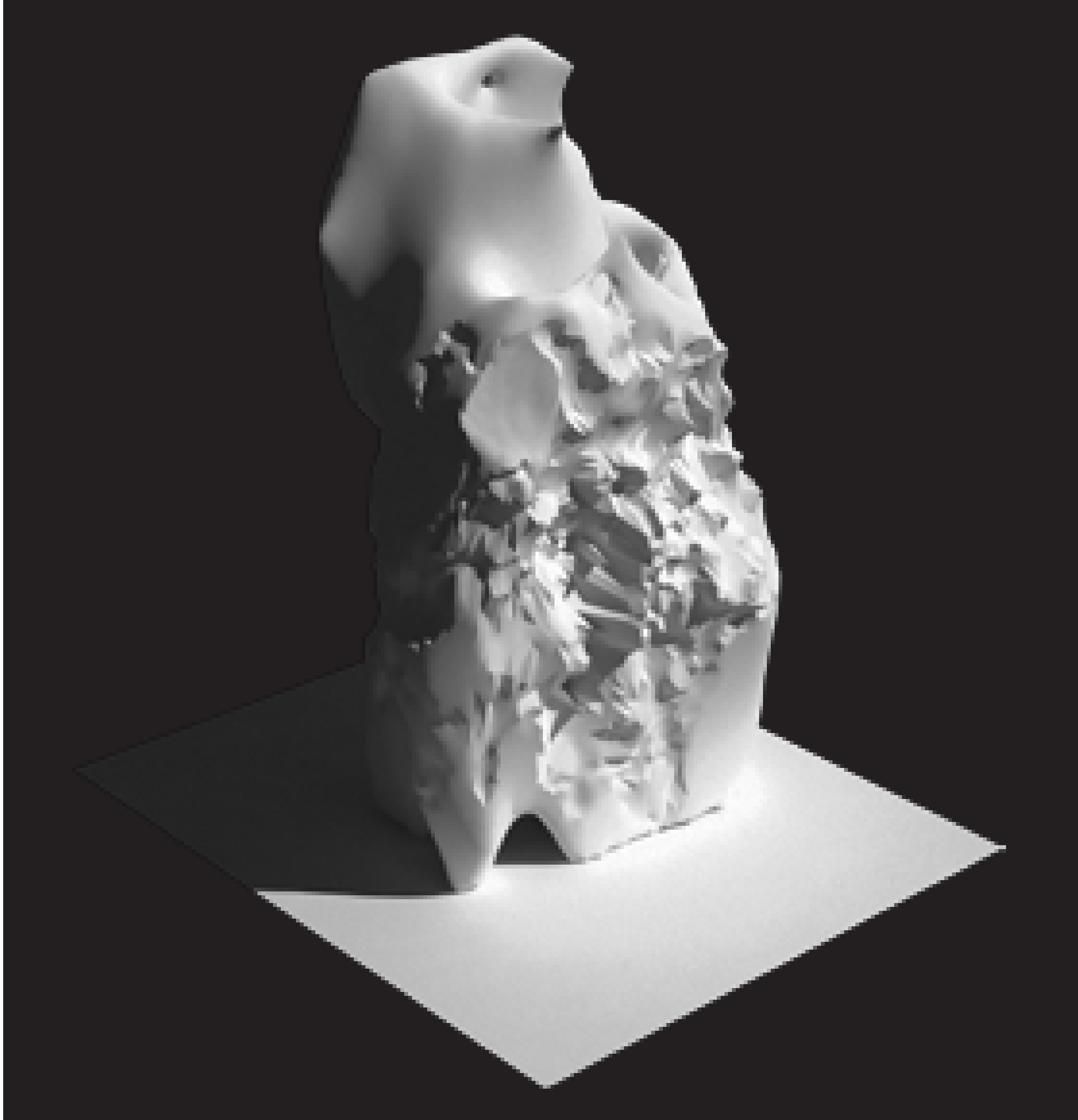
13
SYNTHESIS

In order to use the script creating volumes in conjunction with the one articulating surfaces additional features were needed. The surface articulation tests previously carried out were done on perfect grids and not general volumes and meshes. To reach meaningful results a way for the agents to recognize volumes was needed. This was solved in a computing efficient manner by coding the agents to evaluate wether their future postion were inside the mesh by comparing the degree of the normal and the normal of the closest mesh point. This was roughly 300 times more efficient than my previous solution when using high resolution meshes. Furthermore the possibility to import meshes was implemented, this opened up the possibility of editing in MeshLab or Rhinoceros and to import into Processing again for surface articulation. Subdivision was used to increase the amount of vertice points in the mesh to make the surface behaviour more detailed. The Agent Erosion script was made to only affect points close to Agent Concentrations, this preserves the dynamic of smooth and jagged areas in the surface and prevents visual chaos.

Top: Various stages of surface articulation in progress and a zoom in







Top Left: simple input volume made in Rhinoceros
Bottom Left: isosurface script and subdivision used on input
Right: surface articulation applied

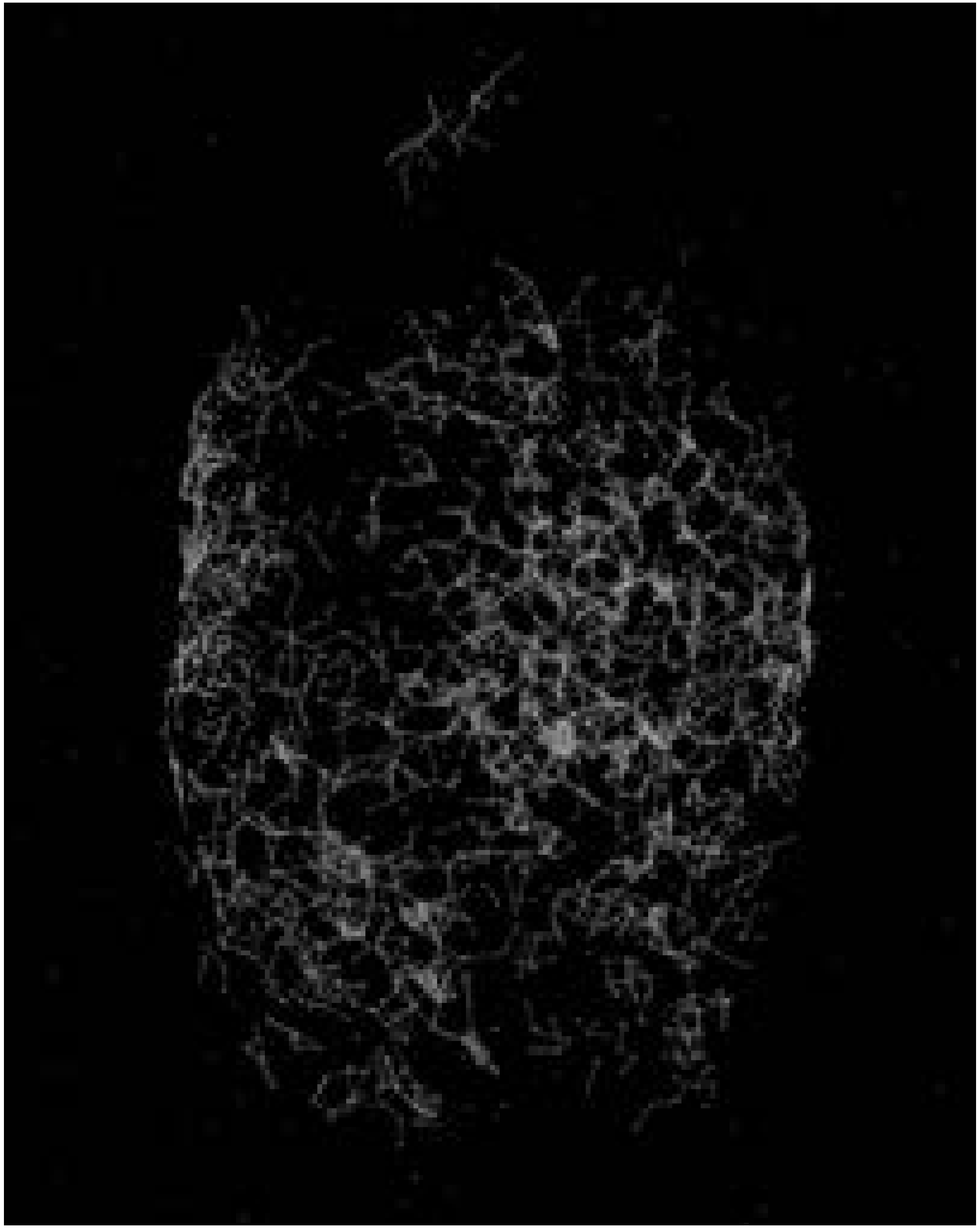
14

OPACITY AND AGENT CONNECTIONS

AGENT PATHS MADE VISIBLE

This technique is a step towards further surface ornamentation, altough not by changing the surface itself but by adding varying levels of opacity and opaqueness. The agents move around an imported volume and freezes when certain conditions are fulfilled. These particles will embed themselves inside the volume and create a sense of depth.

The process begins with a certain number of agents turning into seed points spread out over the surface. More agents freeze in the vicinity of the seed points and grow outwards from there. Lines connect the frozen agents if they are close enough and gives an understanding of the point density.



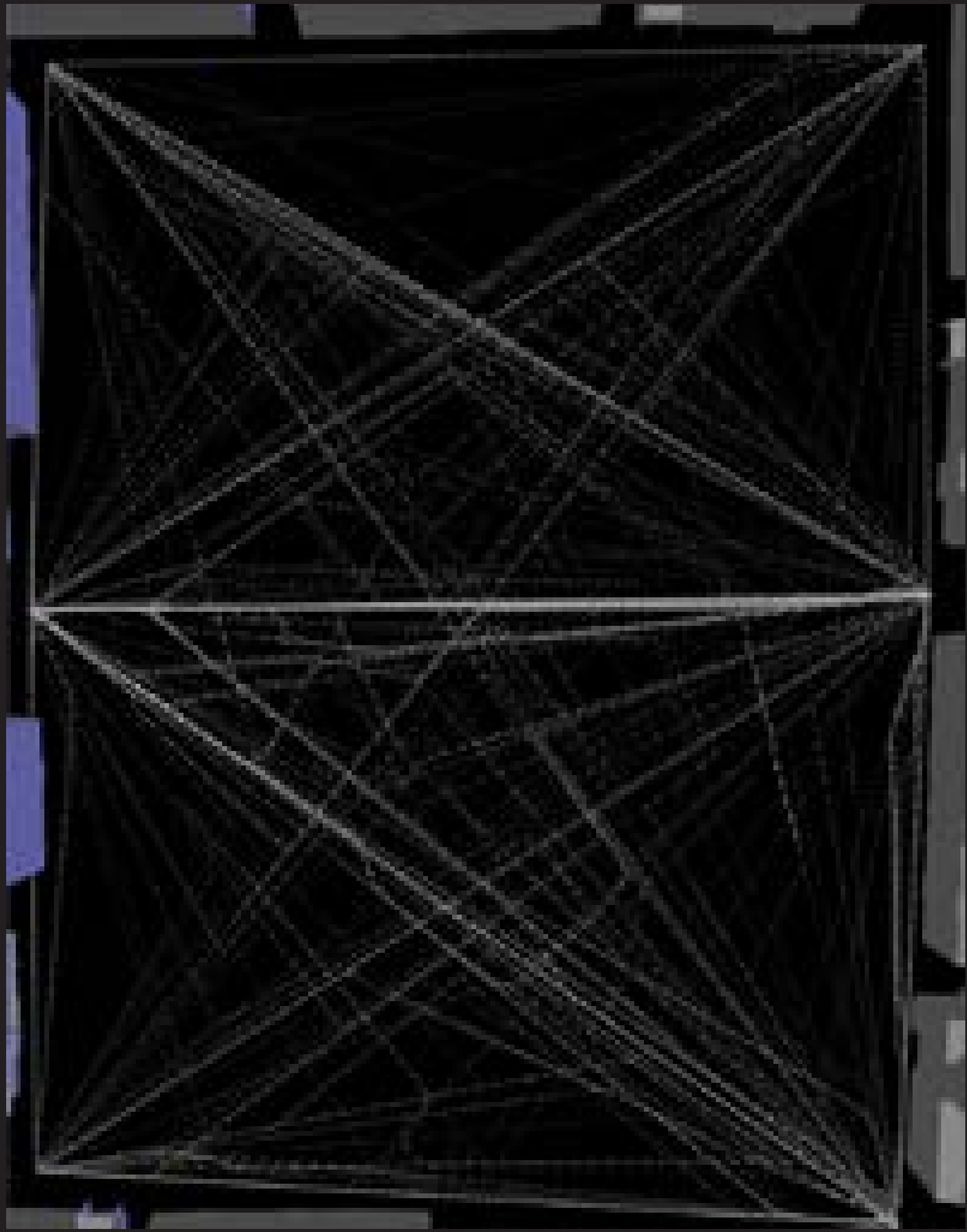
Right: Downward force, 158 seconds elapsed

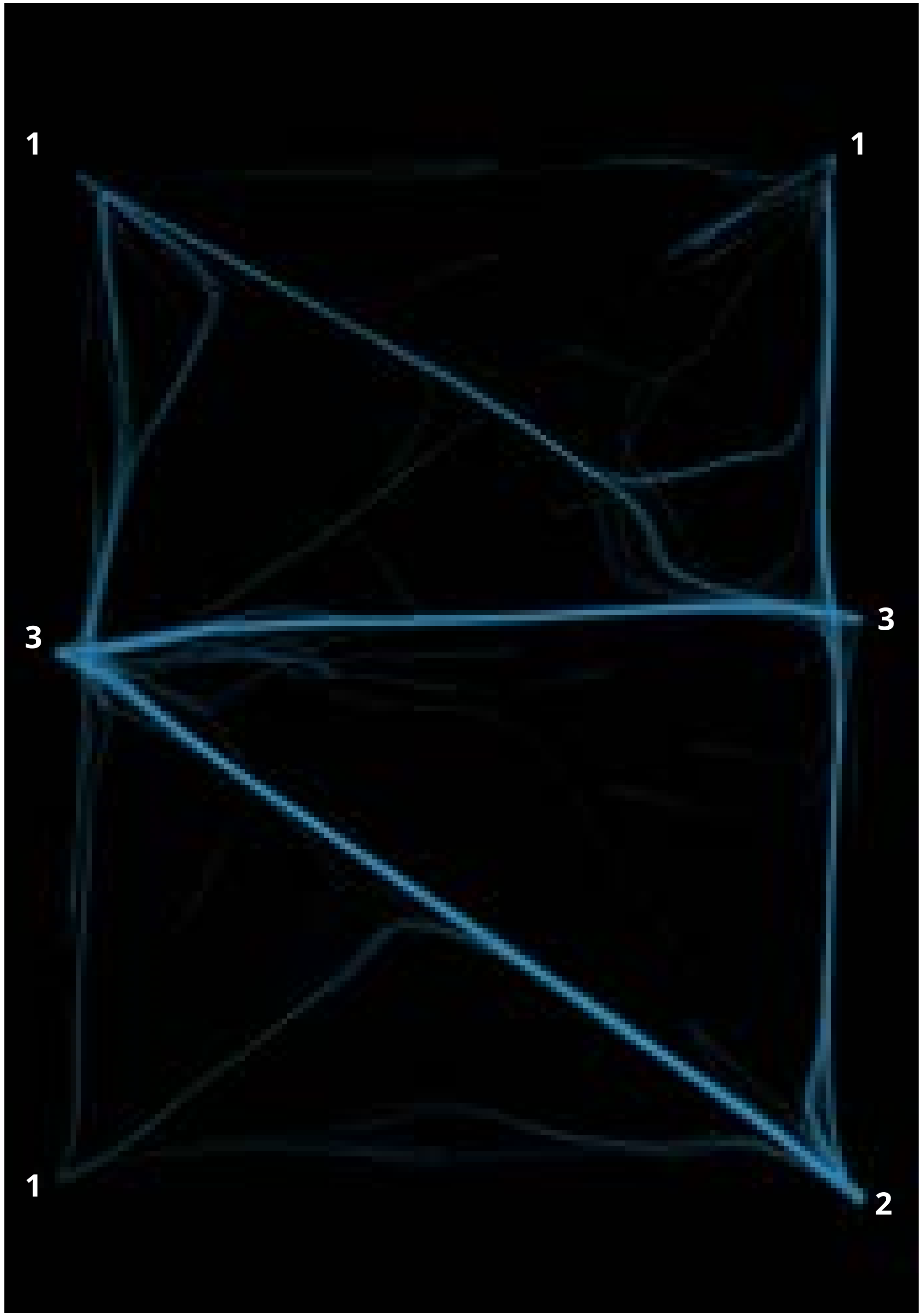
15
Urban Analysis
Stigmergy applied to city

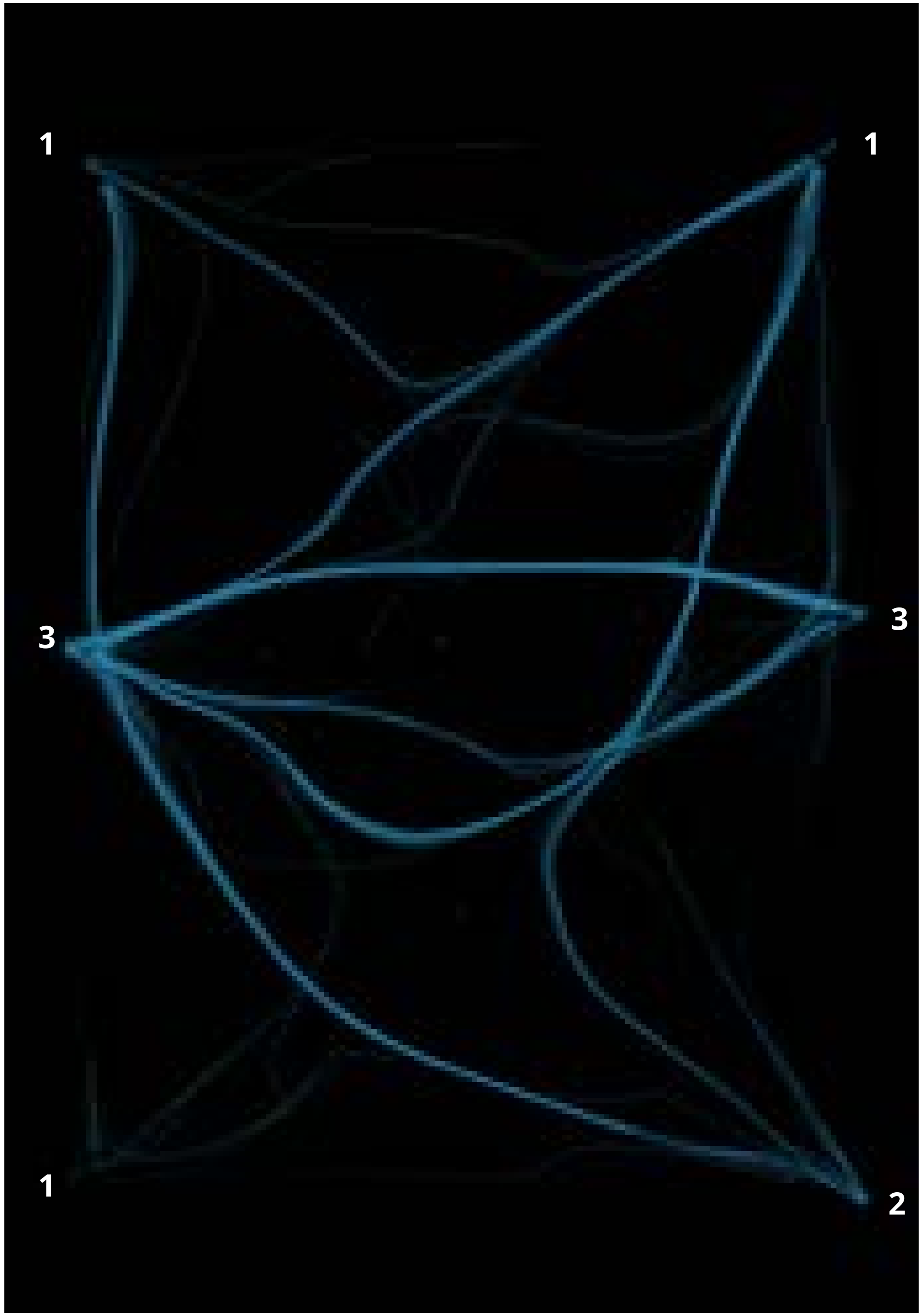
This experiment was made to gauge the agents potential of simulating the movement of people in cities. An open area flanked by connecting streets, attractions and transport nodes were envisioned to see how the agents would navigate between these. Each agent were given a goal to reach and upon finding its target it were given another one at random. At the same time the swarm was imbued with stigmergic behavior, this had the effect of each agent trying to negotiate between two goals: finding its target while staying close to other agent paths.

To see the tendencies of the system clearer all paths were collected as points, point-density is shown through color with warmer colors indicating higher agent activity. Using this aggregated movement strategy made it possible to reveal patterns not visible in the ongoing simulation.

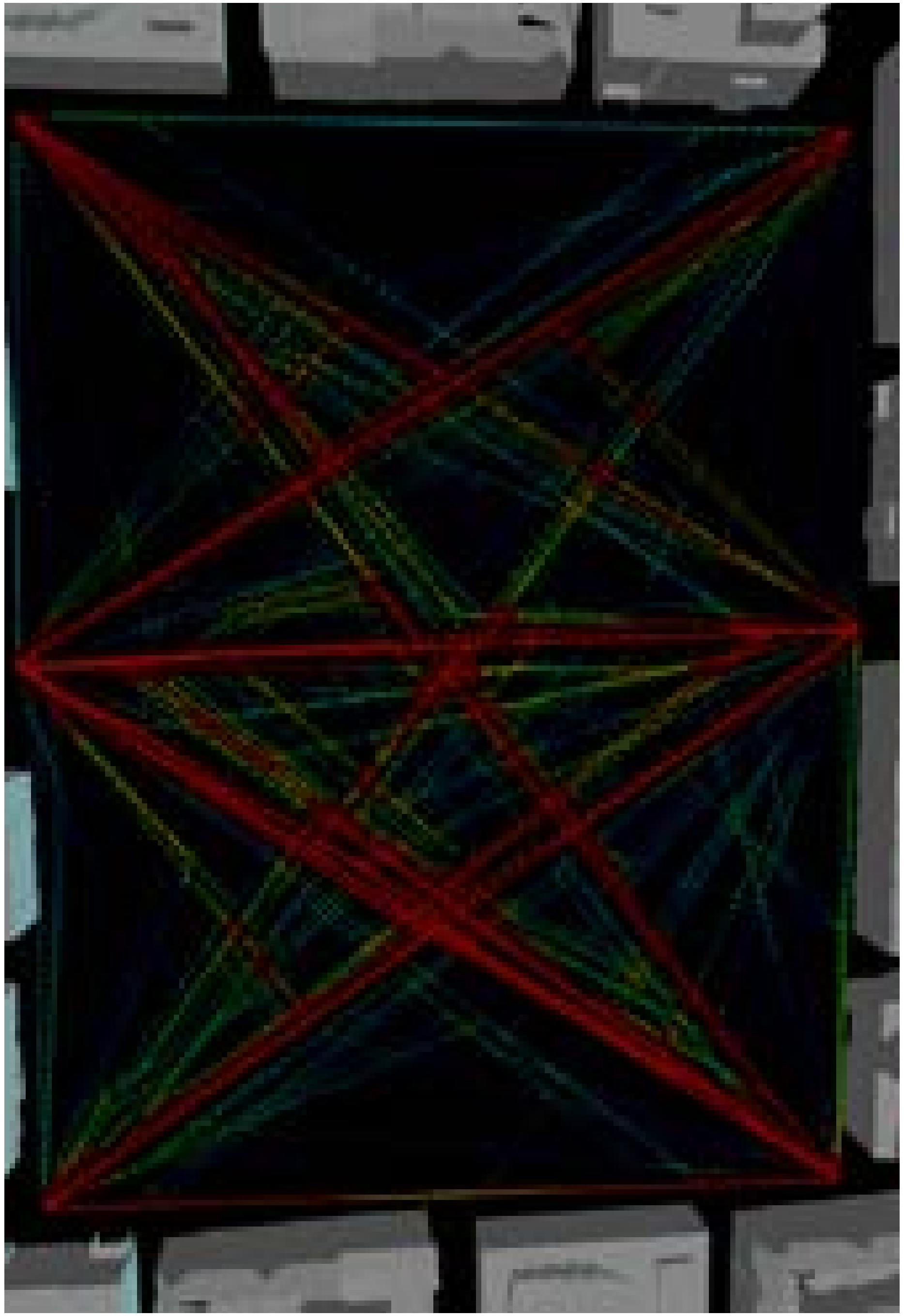
Right: Traces left from agent movement

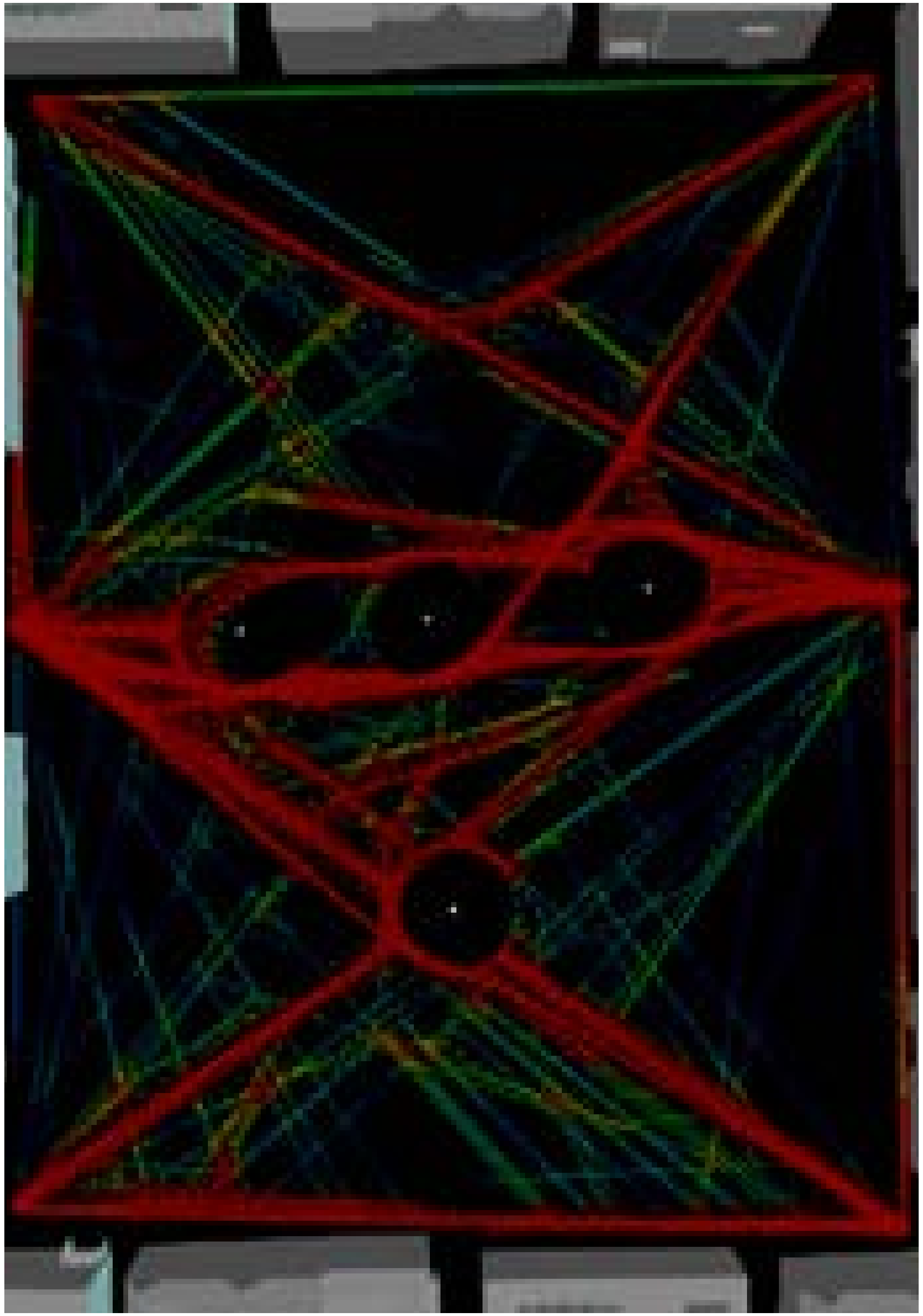




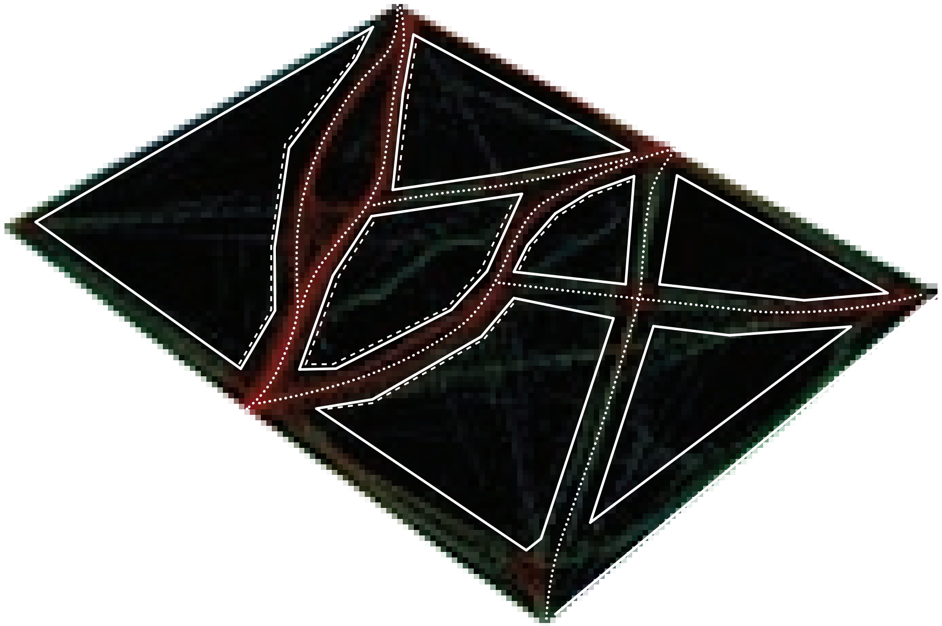


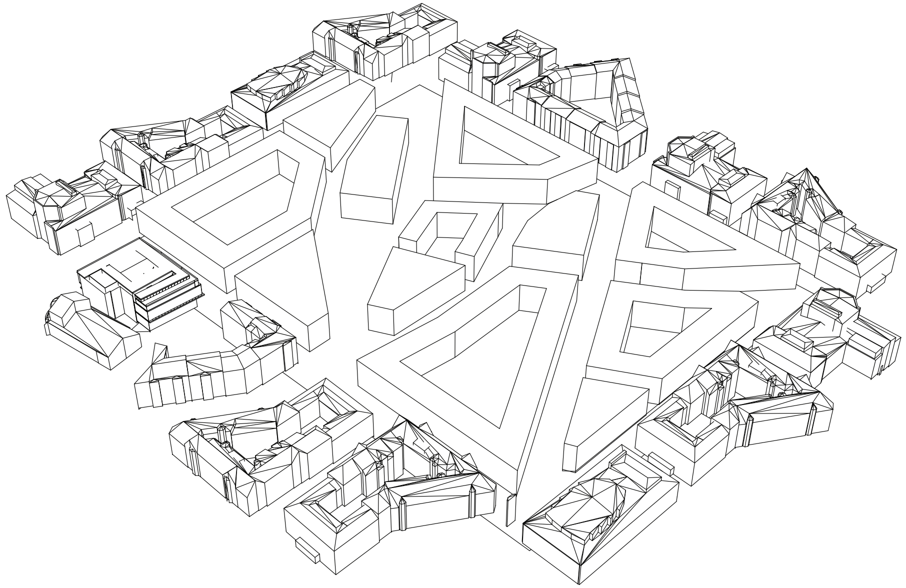
Left: Screenshot of simulation, numbers indicate attractor strength
Right: Screenshot of simulation





Left: Aggregated points from agent movement, warmer colors indicate higher activity
Right: White dots are repellers, the system cope and find new paths





The patterns emerging from the simulations was here used as a basis for an urban layout. High swarm activity dictated how building blocks should orient themselves in terms of active facades.

Each simulation produces a different solution to linking the attractor points together, the tool is best used as way to show different configurations and possibilities.

Left: Active and passive facade locations are identified
Right: Implementation

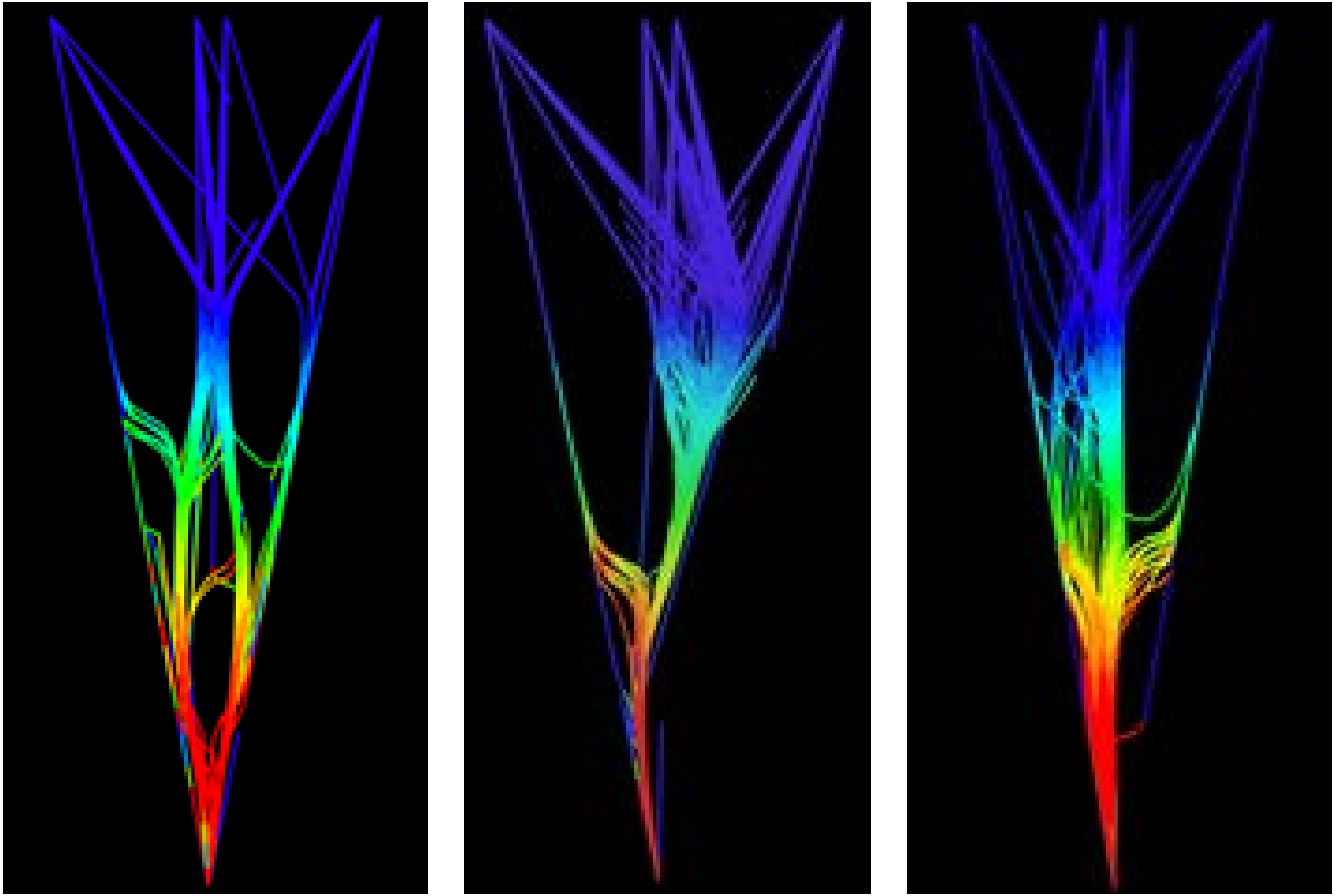
16
THREE TECHNIQUES
Facade implementation

In order to try out the techniques of point generation and surface articulation in an architectural context a facade was devised. In addition to these two methods a third one was created which generated the structural system.

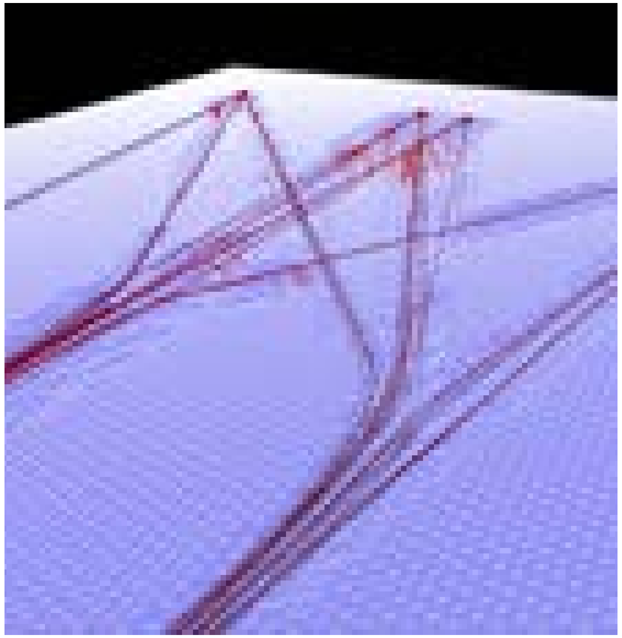
The structural script was based upon the agent erosion code with the attractor logic of the city stigmergy script added. By linking each agents tendency to seek out other agents to its y-position the system could be controlled. In the lower parts the agents tended to stick together while in the top regions they would split up and head directly for the attractors. Over time the script would carve out pillars resembling dendriform structures from a flat mesh.

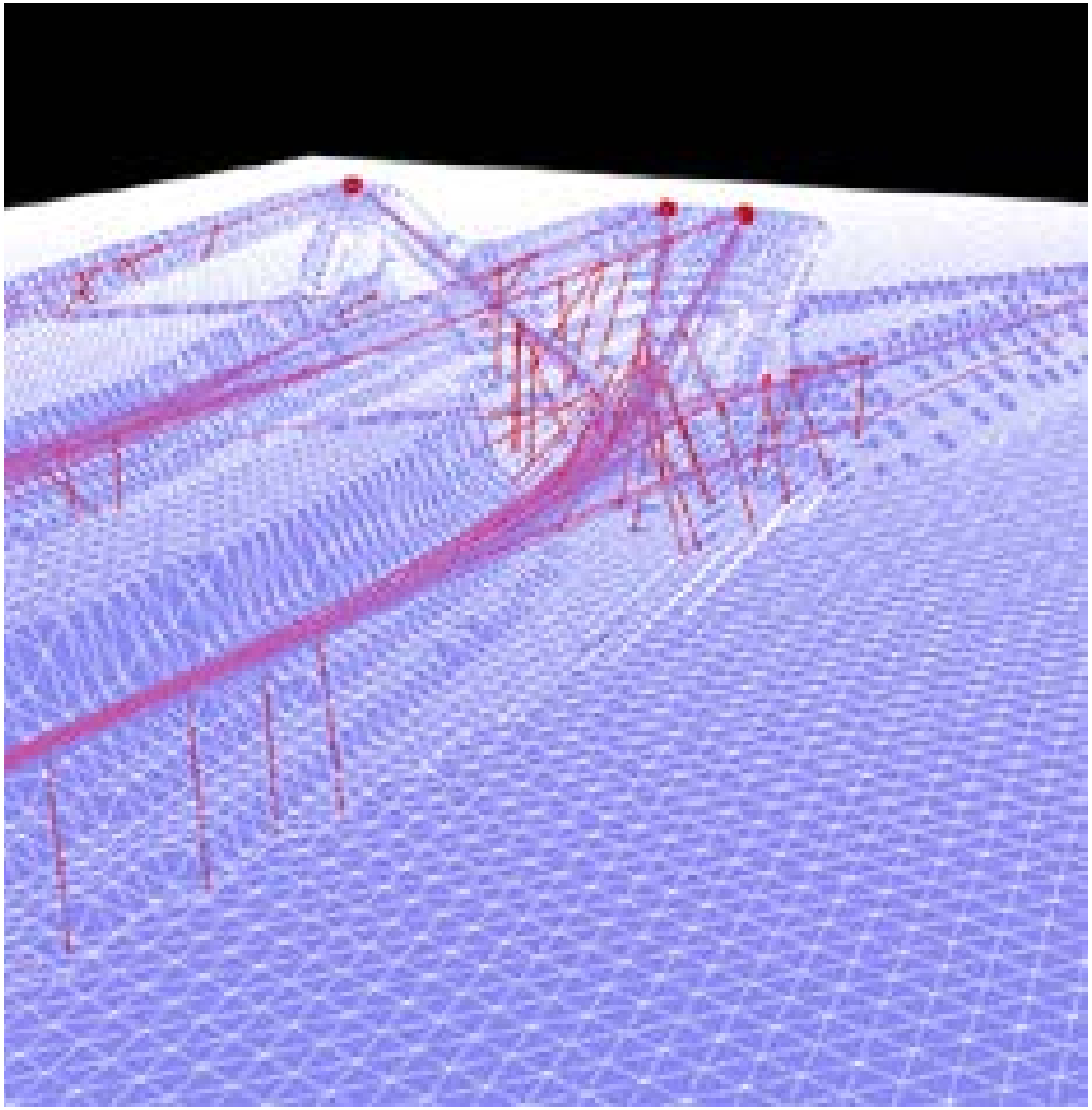
Right: Render of facade

Facing page: Front view of facade with agent paths

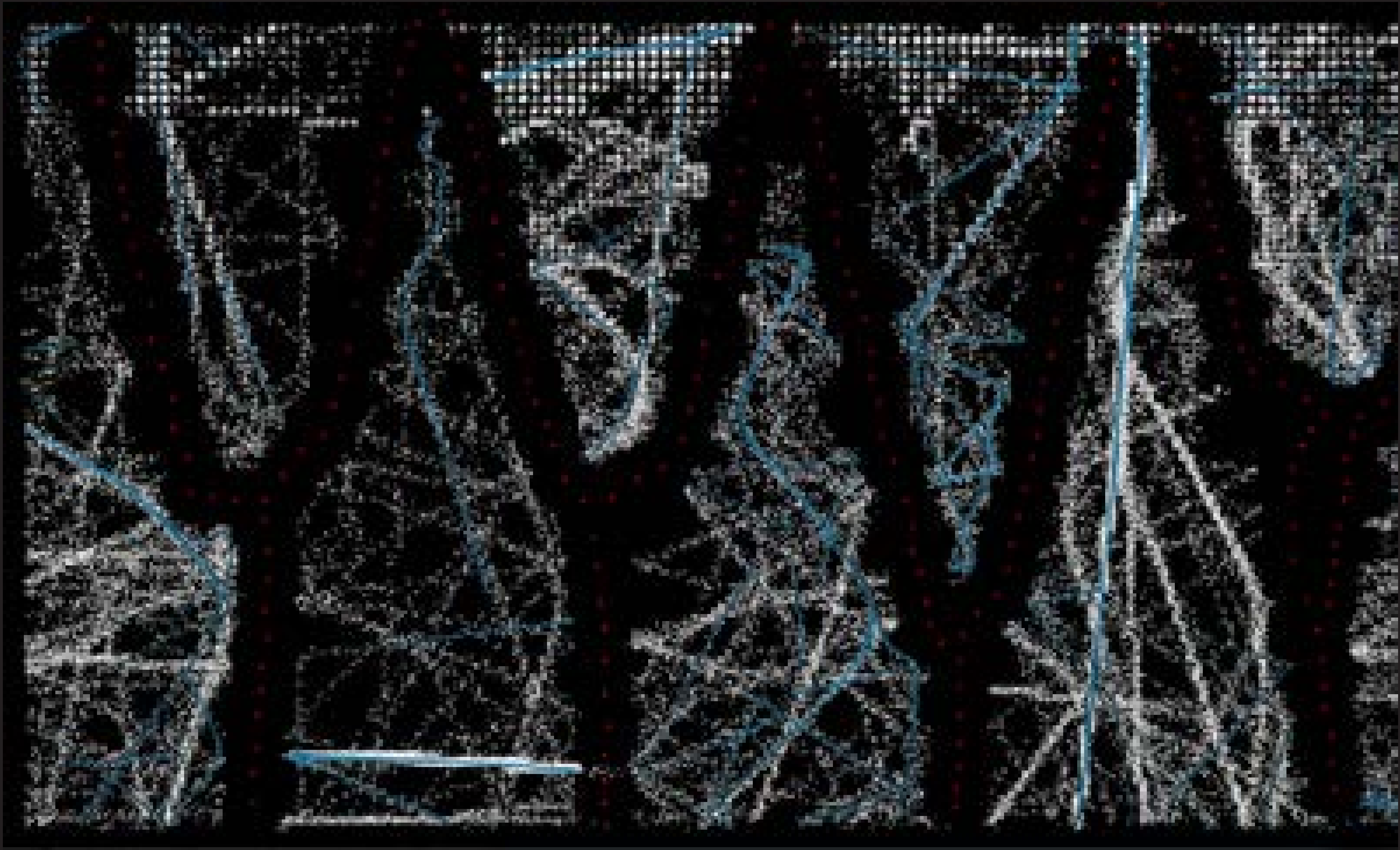


The three images show different simulation results, warmer color indicate higher cohesion value





Screenshots of simulation in progress, agents erode surface to create pillars



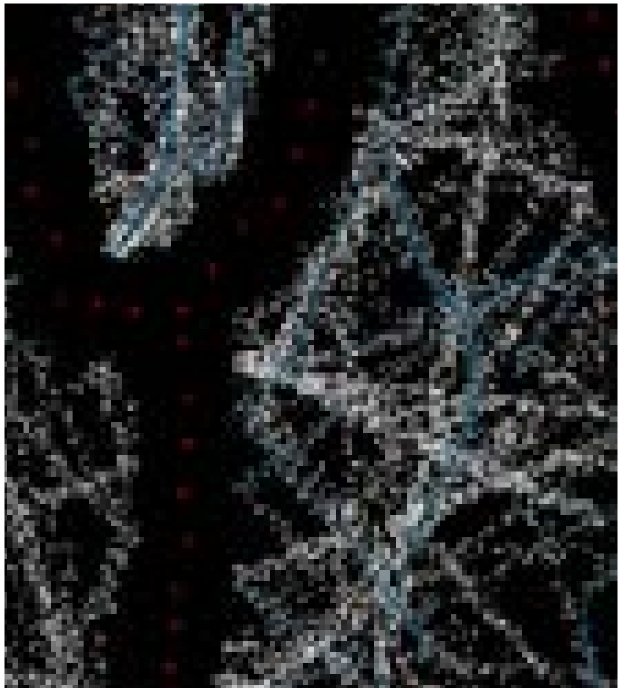
Stigmergic pattern and generated points

PARTICLE GENERATION

Swarms are essentially points which react to each other, a natural progression from this was to device a script around this and to elucidate the agent paths.

The agents move upon a two dimensional mesh surface and avoids repellers that corresponds to the structural system. Each frame the agents identify the closest mesh point and creates a point at the location. By having the points be transparent the movement patterns of the swarm is made visible. By using attractors and repellers the technique can be used to control how much of the interior that is shown according to private and public functions.

The underlying mesh can have a varying resolution which alters the visual appeareance of the particle effect. High resolution produces a smooth and fine grain result while a low quality input tends towards the pixelated aesthetic.



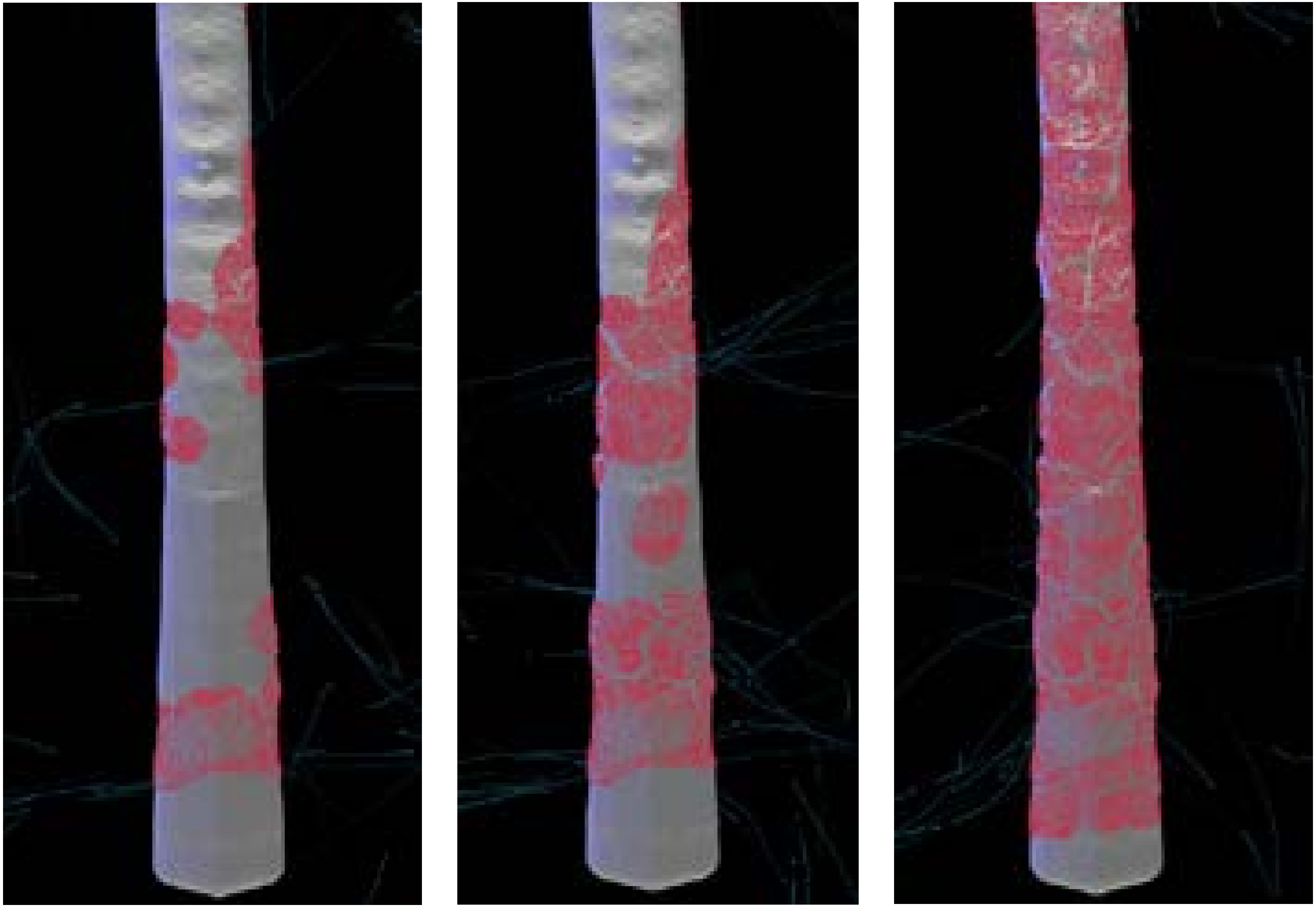
Top: Agent paths and repellers
Bottom Left: Agent paths and particles
Bottom Right: Render showing impact of varying mesh quality

Surface Articulation

Here the technique of surface articulation is applied to the pillars to provide a varying expression and to hook into the glass/resin material. In order to control the process better varying input resolution was introduced in addition to attractors. When the scripts triggers in response to an agent concentration it alters the position of the nearby mesh points. If the affected area is low resolution the effect is barely noticeable. Higher input quality produces flower like patterns with petals directed towards the agent concentration point.

After a mesh point has been moved it is frozen to prevent further movement, otherwise the script would change the input geometry beyond recognition. This also has the effect of creating flower-like patterns with inbetween areas.

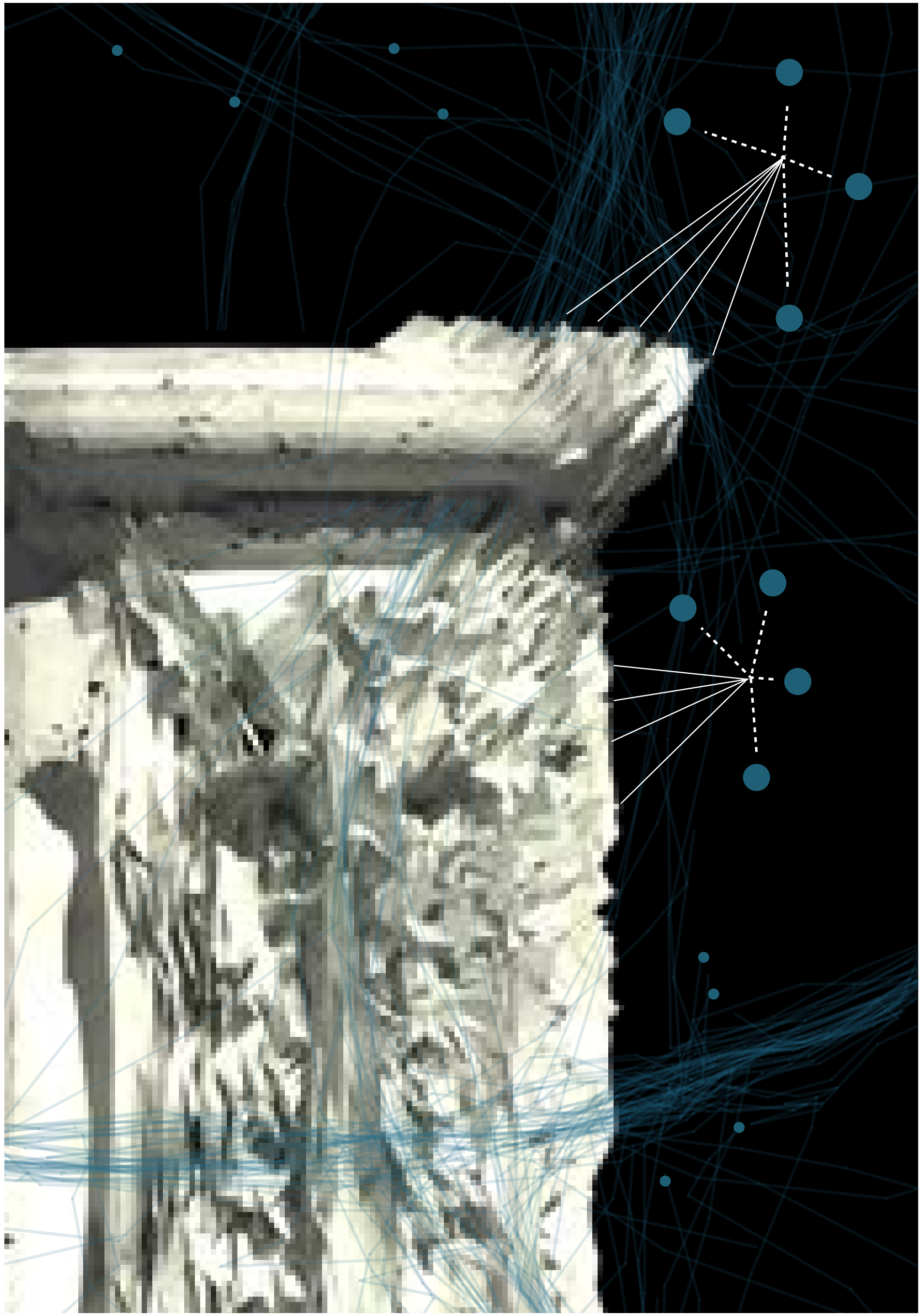
Right: Render of pillar and surface articulation script



Three images showing same simulation 5 seconds apart



Left: Render of facade
Right: 3D-printed part of pillar, flower pattern visible



Left: Script applied to doric column
Right: Close-up

17
REFERENCE LIST

Lynn, G. (2014). Greg Lynn on experimentation, corporate partnerships and making mistakes. *https://archinect.com/features/article/120144801/screen-print-28-place-holder-s-interview-with-greg-lynn-for-issue-1-2-afterschool.* (2018-02-25).

Snooks, R. (2010). # INTERVIEWS /// ROLAND SNOOKS (SWARM 3/3). *https://thefunambulist.net/architectural-projects/interviews-roland-snooks-swarm-33.* (2018-02-20).

Snooks, R. (2014) Behavioral Formation: Multi-Agent Algorithmic Design Strategies

WIscombe. T (2010). INTERVIEW BY OLAF WINKLER. *http://projects.tomwiscombe.com/INTERVIEW-BYOLAF-WINKLER.* (2018-02-20).

//Articles, reference projects and image credits to be added