

# Organic 3D Mesh Creation through Particle-Based Physics Simulation

Sage Jenson

Oberlin College and Conservatory  
jenson.sage@gmail.com

## Abstract

We present a method of generating organic 3D meshes using a particle-based physics simulation. The particles are connected to one another with springs. Each particle's location is recorded at discrete time steps to create a path for that particle. From these paths, triangles are created to form a mesh which is rendered using 3D modeling software.

## Introduction

I became interested in using spring-based physics simulations through electronic music composition [1]. Several of my compositions employed 3D sound fields. In searching for ways to create trajectories to move sound in three dimensions, I settled on methods that use physics simulations. I did this for several reasons: the trajectories sound “natural” because of the physical properties that govern the movement; additionally, generative methods provide huge amounts of material to work with. In building interfaces to create trajectories, I became interested in different ways to visualize the resulting simulations.

The method presented in this paper is an outgrowth of these visualizations. Triangles string together the output of the simulation to create complicated meshes. These meshes are then rendered in 3D modeling software with physically accurate lighting, leading to the illusion that they are sculptural objects. The presented method hints at many options for future exploration.

## Hooke's Law

Hooke's law,  $F = -kx$ , is a simple mathematical model for the motion of a spring. In this equation  $x$  is the displacement between the end of the spring and the spring's resting position, and  $k$  is a constant that specifies the strength of the spring. By Newton's second law,  $F = ma$  (where  $m$  is mass and  $a$  is acceleration, the second derivative of position with respect to time) we have

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x.$$

This is the underlying model that makes the method described in this paper possible.

## Generation of the Mesh

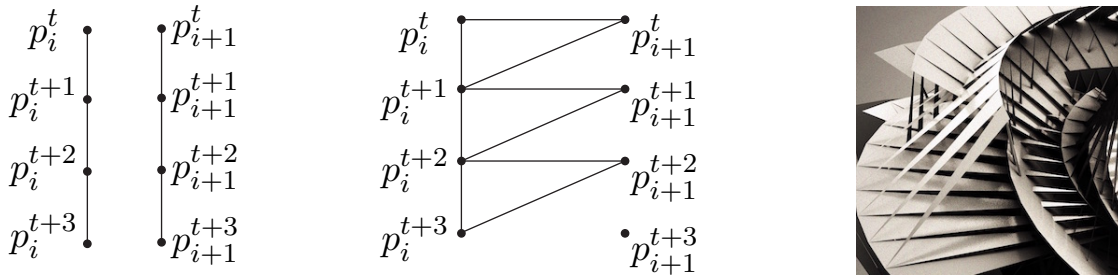
A set of particles  $P$  is created. Springs connect randomly selected pairs of particles. The strength, length, and number of springs are customizable parameters. The simulation involving Hooke's law is run for  $T$  timesteps. Particle  $i$ 's location in timestep  $t$  is notated  $p_i^t$ . The *path* of a particle  $i$  is the sequence of its positions during the simulation, namely  $p_i^1, p_i^2, \dots, p_i^T$  as shown in Figure 1 (left).

The next step is to create a set of triangles that can be interpreted as a mesh by 3D modeling software. The vertices of this set of triangles are particle locations (points in  $\mathbb{R}^3$ ) from the set of paths defined above.

The set of triangles  $T$  that create the mesh is then defined by a set of 3-tuples, each containing particle locations

$$T = \{(p_i^t, p_{i+1}^t, p_i^{t+1}) \mid 1 \leq t < T - 1, 0 \leq i < |P|\}$$

as shown in Figure 1 (center). Figure 1 (right) shows a sample output of a simulation with many particles.



**Figure 1:** Left: The paths of two particles over four timesteps. Middle: The formation of triangles from paths. Right: Sample output.

Various other parameters were manipulated in the simulation. Among these were gravity, friction, the number of springs, spring strength, and spring length as a function of time.

## Implementation

The simulation and construction of the mesh were performed with Processing. Karsten Schmidt's `toxiclibs` Java library [3] was used for the implementation of the physics simulation. Frederik Vanhoutte's `HE_Mesh` Java library [4] handled the construction of the mesh.

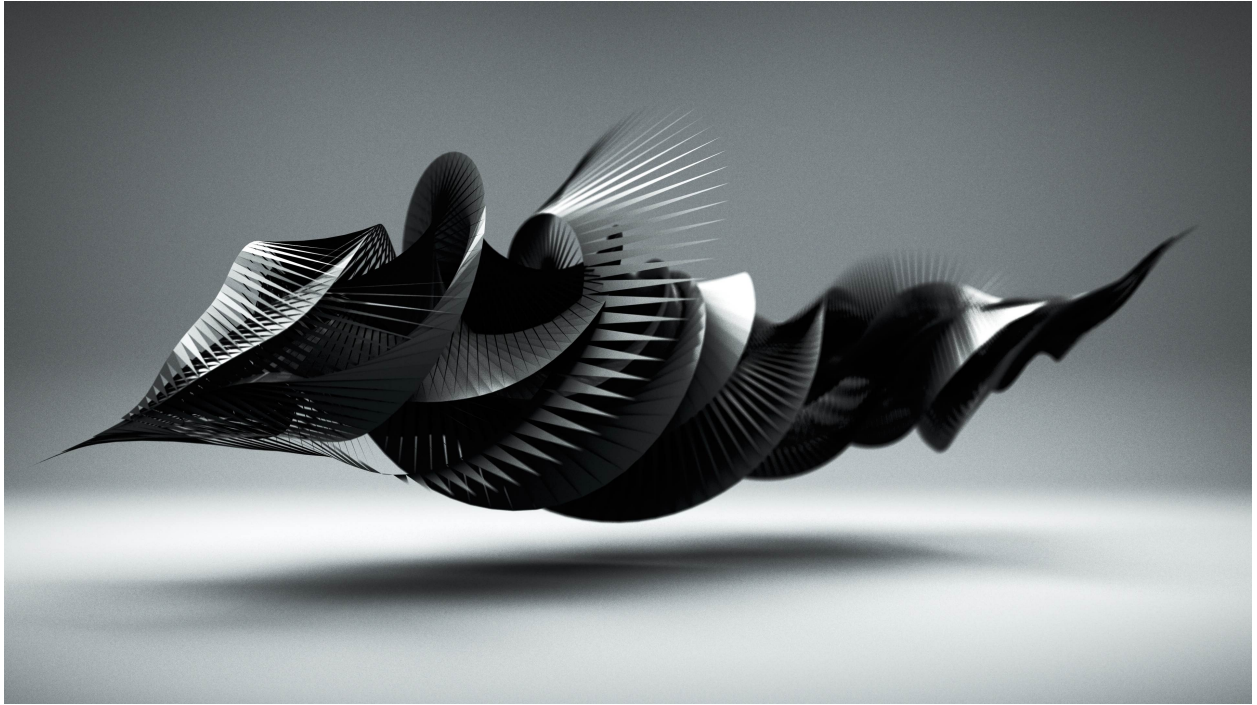
## Rendering of Final Images

MAXON's Cinema4D [2] was used to render Figures 2, 3, and 4 with realistic materials and lighting. Figure 3 was rendered with motion blur in order to make it appear animated.

## Future Directions

There are two main ways in which this method could be expanded: modification of the simulation, and modification of the mesh creation from the paths. The simulation could be modified in many ways, including changing parameters over time, adding new forces, generating more particles at any point during the simulation, etc.

As far as creation of the mesh, the current method forms a collection of triangles that don't have any volume. However one could image alternative methods resulting in objects that have volume, perhaps using tetrahedra instead of triangles. One could also implement more complicated algorithms to turn paths into meshes, such as joining close points together or minimizing the area of the triangles under some set of constraints.



**Figure 2:** *Hummingbird I*



**Figure 3:** *Hummingbird II*



**Figure 4:** *Sheep in the Pen* (Detail)

## References

- [1] Jenson, Sage. (2016) [www.sagejenson.com](http://www.sagejenson.com) (Accessed 1 March 2016)
- [2] MAXON (2016). *Cinema4D* (Version R15.064) [Computer Program]. [www.maxon.net](http://www.maxon.net) (Accessed 1 March 2016)
- [3] Schmidt, Karsten. (2011) *Toxiclibs* (Version 0020) [Source Code]. [www.toxiclibs.org](http://www.toxiclibs.org) (Accessed 1 March 2016)
- [4] Vanhoutte, Frederik. (2016) *HE\_Mesh2014* (Version 3.0.3) [Source Code]. [www.wblut.com](http://www.wblut.com) (Accessed 1 March 2016)