Digital Morphologies: Environmentally-Influenced Generative Forms

Sage Jenson
Oberlin College

Follow this and additional works at: https://digitalcommons.oberlin.edu/honors

Part of the Computer Sciences Commons

Repository Citation
https://digitalcommons.oberlin.edu/honors/200

This Thesis is brought to you for free and open access by the Student Work at Digital Commons at Oberlin. It has been accepted for inclusion in Honors Papers by an authorized administrator of Digital Commons at Oberlin. For more information, please contact megan.mitchell@oberlin.edu.
Digital Morphologies: Environmentally-Influenced Generative Forms

Sage Jenson *
Oberlin College and Conservatory
Advisors: Robert Bosch and Tom Wexler
April 2, 2017

Abstract

We present a generative method to grow triangular meshes with organically-shaped features. Through the application of simplified forces, millions of particles develop into complex 3D forms in silico. These forms interact with external environments in a variety of ways, allowing for the integration of the proposed technique with pre-existing 3D objects and scenes. Large simulation sizes were computationally achieved through the massively parallel capabilities of modern Graphics Processing Units (GPUs).

1. INTRODUCTION

The complexity of Nature is simultaneously aesthetically appealing and an active area of academic study. Such complexity is manifest across many scales, ranging from the macroscopic form of an object to its microscopic structure: contrast the branching structure of a tree to the venation pattern on one of its leaves, or the cellular structure of a single vein. Due to this multi-scale complexity, such organic objects are difficult to model by hand using traditional CAD tools. However, generative techniques can be employed to create the appearance of complexity through the reapplication of simple rules.

This paper presents a method to simulate the evolution of many particles through the application of inter-particle forces. These simulations are hereafter referred to as *growths*. Although growths typically begin with a modest number of particles, each particle has the ability to reproduce by splitting into two child particles. Through modification of the conditions that govern particle reproduction, a variety of spatial forms can be generated.

*sjenson@oberlin.edu
The presented method interacts with pre-existing 3D information (hereafter environments) in several ways. An environment can be used to seed the growth, providing the initial state from which the growth then develops. This methodology can be controlled using the concept of inheritance, allowing artists to integrate the generation of complex, organic patterns into the 3D-modeling workflow. Furthermore, interaction with static environments enables growths to grow on existing environmental features and develop into target shapes.

2. Related Work

Morphogenesis is an umbrella term for the diverse set of processes that transform simple structures into complex organic forms. Also known as artificial evolutionary systems [17], morphogenesis was pioneered by the biologist and mathematician D’Arcy Thompson, who identified many evocative parallels between physical processes and biological forms in his tome On Growth and Form [19]. Thompson postulated that aspects of biological form were dependent on underlying mathematical laws, shifting the object of study from the final emergent form to the process of emergence itself.

Two classical cellular models of morphogenesis are the grammatical and the cell chemistry approaches [17]. The grammatical approach, known as L-systems, was developed in 1968 by Hungarian biologist Astrid Lindenmayer to model the growth of yeast, algae, and other multicellular organisms [9]. Simply defined, an L-system is a context free or context sensitive grammar that works in parallel, iteratively replacing sections of a string using predetermined rules. Lindenmayer realized that the resulting strings can be interpreted spatially, in order to model the processes of cell reproduction and movement. L-systems are widely used to model plants and trees in computer graphics applications because of the relatively simple computational complexity and the variety of forms that can be produced.

L-systems describe a top-down organizational schema that contrasts with the cell chemistry approach, which more closely models biological reality through a focus on bottom-up processes. The cell chemistry approach relies on the reaction-diffusion equations introduced by Turing [21] to simulate the diffusion of one or more chemicals through a connected region of space. In contrast to L-systems, Turing’s reaction-diffusion equations do not actively “evolve” a system; rather they simulate a process on top of a pre-existing topology. Reaction-diffusion equations have been successfully employed to simulate patterns on animal coats, and were used as a nutrient distribution method in previous work [10].

Like the two classical approaches, in this paper we describe a method that seeks to achieve the organic forms found in nature through the application of simple rules. Particles are arranged in a lattice, connected by spring-like links. Forces regulate the spacing of the particles, and often act in competition to create a diversity of structures. We rely heavily on Hart [6] and Lomas [10], who both use a lattice of linked particles, and similar splitting mechanisms for particle reproduction. Lomas in turn cites Kaandorp’s work on modeling accretive coral growth [7]. Because coral is composed of interconnected polyps,
such models naturally translate into linked-particle models. Thus many of the generated forms resemble accretive coral species, most significantly from the genus *Pocillopora* (Figure 1).

![Figure 1: Pocillopora meandrina](image)

In reality, the forces behind the generation of living forms are many orders of magnitude more complex than the presented method. For example, whereas natural objects like coral have intricate interior structures, the presented method is a hollow surface (a 2-manifold). The goal of this work is not to construct a physically accurate model; rather, the presented method seeks to explore how visually complex organic forms can arise from a simplified model: the central problem of morphogenesis.

3. Simulation Mechanics

3.1. Topology and Initial State

A growth is comprised of particles and links, with a variety of forces acting between pairs of both linked and unlinked particles. A growth can be viewed as a mesh with links as edges and particles as vertices. The mesh retains the topology of the initial state. Since the growth is very sensitive to initial conditions, the initial configuration has an enormous impact on the outcome of the process. The implemented initial configurations include an octahedron, an icosahedron, a subdivided icosahedron, a triangulated cuboid of variable height, an isometric grid, and an arbitrary triangular mesh (Figure 2).

3.2. Forces

There are a variety of forces that influence the development of the growth over time, enabling the development of complex structures. The forces determine the next location of the particle for each timestep of the growth. Additionally, the magnitude of each force is scaled by a constant to ensure proportionate interference between forces. The four
main forces described below were modeled after Lomas [10]: the spring, planar, bulge, and collision forces.

A growth is comprised of a set of particles $P$. Each particle possesses a position $p_i \in \mathbb{R}^3$, a normal $n_i \in \mathbb{R}^3$, and a set of linked particles $\mathcal{L}_i$. Of the four main forces, the spring, planar, and bulge forces rely solely on linked particles, and thus are relatively cheap to compute. However, the collision force acts between any pair of particles in close proximity, requiring a more sensitive algorithmic treatment. Notationally, for the vector $x$, $\hat{x}$ denotes the normalized vector, $\frac{x}{||x||}$. To simplify indexing, the following equations describe the forces acting on a single, subscript-less particle $P$ with position $p$, normal $n$, and set of linked particles $\mathcal{L}$.

1. The spring force acts as a linear spring aiming to maintain a fixed distance $S$ between linked cells. Every timestep the displacement due to the spring force is calculated as the average of these linear springs:

$$d_{\text{spring}} = \frac{S}{|\mathcal{L}|} \sum_{i \in \mathcal{L}} (\hat{p}_i - \hat{p}).$$

2. The planar force pushes the particle towards the average position of its linked neighbors, encouraging the mesh to return to a locally planar state. The technique is reminiscent of Laplacian smoothing, an algorithm that moves all vertices to the
average of their respective neighbors 4:

\[ d_{\text{planar}} = \frac{1}{|\mathcal{L}|} \sum_{i \in \mathcal{L}} p_i. \]

3. The bulge force pushes the particle out in the direction of the normal when linked particles are closer than \( S \), the link length. The magnitude of the bulge force is how far, on average, \( P \) would have to move along the normal to allow the links to return to an uncompressed state. For each linked particle \( i \in \mathcal{L} \), let \( \theta \) be defined as the angle between points \( p, p_i, \) and \( t \); where \( t \) is the point \( S \) distance from \( p_i \) along \( n \):

\[ d_{\text{bulge}} = n \frac{1}{|\mathcal{L}|} \sum_{i \in \mathcal{L}} \sqrt{\max(S^2 + ||p - p_i||^2 - 2S||p_i - p|| \cos \theta, 0)}. \]

4. The collision force repels physically-close particles to avoid intersection, acting on pairs of unlinked particles that are closer than a fixed radius \( R \). The magnitude of the force is proportional to the average of the inverse square of the distance. Let \( \mathcal{C} := \{i \text{ such that particle } P_i \text{ is not linked to } P, \text{ yet } ||p - p_i|| < R\} \), that is colliding, yet unlinked cells. Now define

\[ d_{\text{collision}} = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \frac{R^2 - ||p - p_i||^2}{R^2}. \]

To calculate the next position for a particle, each of these forces are calculated and multiplied by variable scalars:

\[ p_{\text{new}} = p_{\text{old}} + c_1 d_{\text{spring}} + c_2 d_{\text{planar}} + c_3 d_{\text{bulge}} + c_4 d_{\text{collision}}. \]

3.3. Nutrients and Splitting

The mechanism for reproduction of particles within a growth is simplified mitosis, or splitting of the particles. When a particle splits, a cleavage plane is first determined by two diametrically opposed linked particles, and the links are split between the parent and the child particles (Figure 3).

The selection of the axis of split has a significant impact on the final state of the growth. Two methods are implemented to determine the axis of split: the trivial method (simply select the first link and the respective opposite for the axis) and the shortest such axis. In Figure 3, the axis of split used is between particles 1 and 4, the shortest such axis of split. A visual comparison of the reapplication of the two methods can be found in Figure 4.

Each particle \( i \) has an internal nutrient level \( n_i \). A particle splits when \( n_i \) is above a constant threshold \( T \). Each timestep a variable amount of nutrient is added to every cell. From the methods below, 1 and 2 are from Lomas [10].
1. A constant amount.
2. A random amount for each cell.
3. An amount proportional to the curvature of the mesh at the particle.
4. An amount proportional to the number of particles within $R$ of the particle.
5. An amount proportional on the age of the cell.
6. An amount proportional to the area of the faces.
7. An amount derived from the position of the cell.
8. An amount derived from other environmental parameters.

The above splitting method relies on the existence of a complete one-ring neighborhood of each particle. However, such a neighborhood does not always exist (e.g., the edges of a certain starting configurations). In these scenarios, the particle is marked frozen. In the growth, frozen particles do not update their position or split, yet still can exert forces on the cells around them such as the collision force.

3.4. Environments

LIDAR [22], photogrammetry [12], and other 3D scanning techniques can transform the spatiality of the world into detailed 3D computer models. The term environment is here defined to be any object in the world-space of the growth that is not itself created purely from the generative processes described above. Such objects include textured 3D scans made by the author and various creative commons mesh and texture files (see Appendix A). The growth can interact with such external objects in a variety of ways:

1. Start State: the environment can serve as the initial configuration for a growth.
2. Repulsion: the vertices of the environment apply a collision force on the particles in the growth, similar to the one described in Section 3.2.

3. Anchoring behavior: when a particle gets within a certain radius of the environment it is frozen, attaching to the model. Alternatively, once a particle’s position crosses an environmental boundary, it is frozen.

4. IMPLEMENTATION

Two implementations of the presented technique were built. The first is an application for realtime parameter adjustment on small datasets. This application was built using the openFrameworks C++ library [14], allowing for rapid visual feedback. This feedback facilitated the finding of ideal parameters for a given scenario. The second component of the implementation is a command-line application that can handle very large growth sizes through the use of massively parallel processors.

4.1. COLLISION DETECTION

The most computationally complex operation in the growth is the collision detection step. A brute-force implementation that checks every particle against every other is \(O(n^2)\), making it infeasible on very large growths (even in massively parallel environments). Spatial partitioning provides a substantial speed increase to an average performance of \(O(n \log n)\) [16]. In this algorithm, 3D space is partitioned into boxes that are of size at least \(2R\). Thus for any particle \(p\) in box \(b\), collision tests must be computed only between
and (1) the other particles in $b$ and (2) the particles in $b$’s 26 neighbors, significantly reducing the search space.

4.2. Normal Calculation

To calculate the normal for a given particle $P$ with position $p$, the one-ring neighborhood of linked particles is retrieved. If this neighborhood does not exist (e.g. particles with no links), the particle is marked frozen (as discussed in Section 3.3). Once the one-ring neighborhood is found, the cross products of each sequential pair of neighbors is computed and normalized:

$$n = \sum_{i \in L} (p_i - p) \times (p_{i+1} - p_i).$$

When $i = |L|$, let $p_{i+1}$ be $p_1$ to complete the circuit. Thus far this technique doesn’t account for the orientation of the normal, which depends on the orientation of the one-ring neighborhood. We use the normal from the previous time-step to determine the orientation of the current normal. Thus if $n$ is pointing in the entirely different direction as the previously computed normal (with the dot product $< 0$), then $n$ is scaled by $-1$. This technique relies on the assumption that a normal will remain pointing in the same general direction from one frame to the next.

4.3. GPU

The presented method, like many particle-based simulations, is remarkably parallelizable. For each time-step, every particle need only access previously computed information to calculate the next position.

Oberlin College’s NVIDIA Tesla K40M GPU Computing Accelerator was used to compute the majority of the growths in this paper. NVIDIA’s CUDA platform [13] was a natural choice to implement the project as it allowed for seamless integration with existing C++ code.

4.4. Rendering

Every growth can be translated into a triangular mesh by design. While previous work [6, 10] has rendered similar particle-based simulations using spheres to represent each particle, the triangular mesh allows for smaller file sizes without visually losing spatial information from the growth.

MAXON’s Cinema4D R16 [11] was used to render the majority of images in this paper. Materials and lighting were added, yet the model’s vertices and faces remained consistent with the output of the growth. Meshlab [2] was used extensively for intermediate visualization, and to render Figure 14a.

Because each growth develops over time, renders of each timestep can be assembled to form an animation. The animation can give insight into the generative structure of the growth. Several animations from this project can be found at [http://bit.ly/2nxqBFd].
4.5. Code

The majority of the project was written in C++ using the CUDA platform [13].

The Simulati... the main algorithmic loop for each frame. It relies on many other classes that perform specialized functions.

The Particle class contains particle information including position, normal, link information, nutrient level, age, number of collisions from the last frame, and box index in the grid. It contains methods to compute the next position.

The Parameters struct contains the information that the parameter information that the Simulation needs.

Octahedron, Icosahedron, Plane, Tube, Multiple, and OBJModel all implement the abstract class InitialState. These files contain information for the initial position, normals, and topology of the growth, and the environment if specified.

The Exporter class handles the exporting of OBJ-formatted 3D models from the growth. It requires a Parameters struct to include growth parameters as comments in the exported OBJ file.

5. Results

Clearly, fully exploring the parameter space described above is combinatorially prohibitive. The results presented here represent the beginning of an effort to delineate the search space, seeking to build intuition about the emergent behavior of the system while exploring how certain formal goals may be met. A comprehensive list of growth parameters, including growth size and timestep, can be found in Appendix B.

Section 5.1 examines the effect of initial start states on the output of the simulation, from simple topologies such as an octahedron, to more complicated triangular meshes. Section 5.2 investigates nutrient distribution schema, beginning with simple spatial techniques and progressing to trait inheritance and curvature calculation. Section 5.3 concludes by demonstrating the possibilities for interaction with external 3D environments.

5.1. Start States

5.1.1. Regular Initial States: Octahedron and Icosahedron

The octahedron and icosahedron start states provide several advantages. Since both are Platonic solids, particles in growths initialized from these states are identical: from number of links to spring, planar, bulge, and collision forces. Thus the complexity arising from these start states is purely due to the growth’s process, as opposed to differences between initial particle configurations. By using such a start state, the different results from area-based verses random nutrient distribution are visually clear (Figure 5). Area-based distribution results in a surface with relatively regularly sized features, while the random distribution results in several thin, irregularly shaped strands extending from the edges of the growth. This artifact is present in many subsequent growths, thus warrants a brief discussion.
Thin strands emerge from a variety of initial states and nutrient distribution. This artifact arises primarily as a result of the collision force paired with favorable particle-link configurations. Because there is no collision force between linked particles, linked particles can group together in tight configurations, as can be seen in Figure 6. It is clear from Figure 7 that the strands are capable of self-propagating; that is once a strand develops, that strand continues to grow while retaining a thin structure.

5.1.2. Cuboid Initial State

The initial cuboid state is incredibly simple (see Figure 2c). However, a diverse set of forms can be generated through varying initial parameters. Figure 7 illustrates the progression of area-based distribution from a cuboid start state with a height of 101 particles. Area-based distribution allows for structural variation within a growth: the ends of the
cuboid develop spherical nodules, while the connective tissue in-between remains a very thin strand. This variation highlights the sensitivity of area-based nutrient distribution to initial conditions. The thin strand can be eradicated through an increase in the bulge-force coefficient, which results in a relatively uniform thickness throughout the length of the growth (Figure 8). Additionally, the repeating structure of the cuboid results in repeating sections throughout the growth, evocative of skeletal patterns and insects. Unlike the other examples which employ longest axis-of-split, Figure 9 demonstrates the results of the trivial axis-of-split on the cuboid initial state.

5.1.3. Mesh Start State

A start state need not be a simple shape like an icosahedron or cuboid: any triangular mesh can be used as input for the growth, seeding subsequent development. This has the effect of developing pre-existing features of the mesh, as can be seen in Figure 10 seeded with the child mesh (far left). Notice how even after 150 time-steps, features such as the ears and nose result in uniquely sized features. The variation is further developed in Section 5.2.3.

5.2. Nutrient Distribution Variations

5.2.1. Spatial Nutrient Distribution

Nutrients need not be distributed uniformly throughout the growth. Implemented variations on spatial distribution include nutrients allocated proportional to:
(a) Frame 18 with 18674 particles.

(b) Frame 33 with 76,471 particles.

(c) Frame 74 with 809,890 particles.

Figure 8: Similar to Figure 7 yet with a larger bulge-force coefficient. Displayed to scale.

Figure 9: Progression with area-based reproduction and the trivial axis-of-split.

Figure 10: Simulation seeded with a mesh using area-based nutrient distribution.
1. The $x$ position of the cell (Figure 11a).
2. The distance of a particle from the origin (Figure 11b).
3. The square distance of a particle from the origin (Figure 11c).

The isometric grid start state allows for an intuitive understanding of the outcome of the above techniques. This is particularly apparent in Figure 11a, where the gradient result clearly illustrates the underlying mechanics. In Figures 11b and 11c, multiple sphere-like levels of the growth are pushed outwards because of the extremely high levels of reproduction at the center of the growth. Linear distance distribution creates a varied outer layer, while the square distance distribution creates a relatively low-resolution outer layer.

5.2.2. Age-based Nutrient Distribution

Recall that each particle $i$ has an age $a_i$. For particles that are part of the initial state, the age is the same as the time-step of the growth. For particles that are created through splitting, the age is simply the time since their parent split. Let the lifespan of a particle,
notated \( l \), be defined as the amount of time after a cell is created that a cell receives nutrients. The amount of nutrients distributed to a particle of age \( a \) is defined as \( n_a \) and given by the piecewise linear function

\[
n_a = \max\{l - \|l - a\|, 0\}.
\]

This has the effect of only adding nutrients during the particle’s lifespan, with the maximum allotment of nutrients at the middle of the particle’s life. After a particle’s age surpasses its lifespan, the particle doesn’t receive any more nutrients. Figure 12a demonstrates the effects of purely age on an icosahedral start state. Although this cannot be seen through still images, the growth progresses in regular waves, coinciding with the magnitude of the age function given above.

### 5.2.3. Curvature Nutrient Distribution

Curvature is widely used in computer graphics applications, from remeshing [11] to non-photorealistic rendering [8]. While more accurate estimations of curvature over a triangular mesh exist [15], the approximation used for the application of the paper relates the normal of a particle to the average position of the linked particles. For a particle \( P \) with position \( p \) and normal \( n \), let the average position of \( P \)’s linked particle’s be denoted \( q \), and let \( r = q - p \). We now define the curvature to be

\[
c = ||r|| \left( \max\{-(n \cdot \hat{r}), 0\} \right).
\]

The dot product of the \( r \) with \( n \) is a scalar that can be interpreted as determining whether \( P \) is bulging outwards or inwards. The curvature as defined above can be added
to the growth’s mechanics with minimal computational cost. This definition of curvature is unique to the presented application, and should not be confused with the many other definitions of curvature.

For each particle at each time step, \( c \) is calculated and added to the particle’s internal nutrient level. An octahedron (Figure 13) and the child mesh (Figure 14b) are used as the initial state for this growth. The color map in Figure 14a visually renders the curvature from a time step of the growth. Unlike previously explored methods, curvature-based nutrient distribution results in parts of the growth moving between both thin strands and more voluminous sections.

5.2.4. Inheritance

Inheritance is here defined as a child cell receiving certain traits from its parent. These traits could be any particle parameter—spring length, collision radius, etc. By holding such traits constant throughout the growth, the initial state is able to maintain a significant presence over time. The implemented method integrates inheritance with curvature-based nutrient distribution from section 5.2.3. At the onset of the growth, every particle calculates its initial curvature, a value which is saved and passed on to its children. Nutrients are distributed to the particles proportional to this initial curvature. Figure 15 demonstrates how curvature-based inheritance can selectively develop sections of a mesh seed. An exciting corollary of the above method is that an artist can define where and how much they desire a growth to develop, simply through assigning different
weights to the set of vertices.

5.2.5. Buds

To simulate tentacle-like extremities on a growth, reproduction can be limited to a small subset of the cells (after Hart [6]). Such cells are labeled buds, so called because they behave like the growing end of a plant. Nutrients are added to these cells every time step, causing them to exhibit very fast growth which in turn results in the creation of a strand of particles in its path. In order to create extremities with increased width, non-bud cells are allowed to split $K$ number of times. This causes areas near the buds to expand, widening the extremity. Additionally, the axis of split has a significant effect on the shape of the extremities. Figures 16a, 16b, and 16c demonstrate longest axis-of-split for $K \in \{0, 1, 2\}$. Figures 16d, 16e, and 16f demonstrate trivial axis-of-split. Unlike the other parameter combinations, using trivial axis-of-split with an $K$ value of 1 results in
the growth of consistently shaped extremities (Figure 16e).

Branching can be achieved by adding buds over the course of the growth. To create a fork in a branch, a single one of an existing bud’s newly formed children is designated a bud. As a result, the old bud continues creating an extremity, while the new bud creates its own extremity as can be seen in Figure 17a. Figure 17b is the same growth a considerable number of time-steps later.

Figure 16: Budding behavior with longest (a, b, c) and trivial (d, e, f) axis-of-split.
5.2.6. Hybrid Methods

The aforementioned nutrient distribution variations can be combined and layered to create growths with multiple types of features. We present two examples: Figure 18a uses inherited curvature values to determine the distribution of nutrients for all initial points that are above a certain curvature threshold, while the rest of the particles subscribe to area-based nutrient distribution. Figure 18b uses a sinusoidal function of a particle’s position to interpolate between curvature and area-based nutrient distribution.

5.3. Interaction and Environments

5.3.1. Coexistence

It is possible to create an arbitrary number of individual connected components that coexist in the same space, using only the methods described above (Figure 19). The collision force is the only arbiter between individual connected components. Depending on the initial configuration, the interaction between two connected components in close proximity can create squished surfaces with different visual characteristics than the sections of the component that are not in close proximity. The squished surfaces have a higher density of points, an artifact from the relatively high density of particles.

5.3.2. Repulsive and Freezing Environments

Interaction with static environments can take a variety of forms. Two contrasting approaches are repulsion and anchoring. Repulsion is an intuitive extension of Section 5.3.1.
Figure 18: Hybrid nutrient distribution.

(a) Two connected components, with two different perspectives per component.

(b) Three connected components, with multiple perspectives on the middle component.

Figure 19: Two and three connected components.
where the points that comprise the environment act exactly like a connected body that does not move, yet still exerts collision force on other particles. Such a technique seems to work well in cases where the environmental points are relatively dense and uniform, such as the torus environment in Figure 20. Unfortunately, most meshes do not have dense, uniform point distribution. Figure 21a demonstrates repulsive interaction with an environment with sub-optimal point distribution. As can be seen in the underside view in Figure 21b, it is possible for the growth to pass through the surface of a repulsive environment if there is adequate force being exerted from the side of the growth.

To address this problem, an alternative method of interaction between a growth and the environment is used. Namely, when a particle $P$ in the growth is within a certain radius of an environment particle, $P$ is frozen (i.e. neither splitting nor moving from that point onward). Figures 21c and 21d demonstrate such a mechanic. However, there remains space between the environment and the growth (Figure 21f). To create a tighter fit between the growth and the environment, the freezing force is only applied if a particle travels “through” an environmental particle. Specifically, for a particle $P$ with position $p$ and normal $n$, and a set of points that comprise the environment $E$, let $e \in E$ with normal $n_e$ be the closest point within distance $R$ of $p$. If no such $e$ exists, do nothing. Otherwise, if

$$ (p - e_n) \cdot n_e < 0, $$

freeze particle $P$. As can be seen in Figures 21e and 21g, a much closer fit between the environment and the model is achieved.

The form of flora is incredibly dependent on external forces such as air movement, water flow, and gravity. A constant directional force can allow for a simple imitation of these complex forces, allowing for the appearance of more natural integration within an environment. In particular, when the force is applied in the downwards direction, it simulates a gravitational force (Figure 22). Figure 22a maintains the same parameters (except for the gravity force) as the growths in Figure 21, whereas Figure 22b substantially increases the strength of the bulge force.
Figure 21: Environmental interaction.

Figure 22: Gravity and environments.
5.3.3. Target Shapes

A natural extension of growth around an environment is growth from within an environment. If a growth is adequately contained by an exterior environment, with the right parameters it will take on the shape of its container. However, when repulsive methods are used as described in Section 5.3.2, similar challenges arise from the growth breaking through the container (Figure 23a). Although such an artifact may be artistically provocative, it does not achieve the goal of creating the desired shape. Freezing particles that escape through the mesh can adequately address this problem, as proposed in Section 5.3.2. Figure 23b demonstrates the referential potential of such a technique. The snout appears distorted because the inside of the dog’s teeth collide with the growth.

6. Conclusion and Future Work

The presented method demonstrates a subset of the innate morphological possibilities that arise from a relatively simple set of constraints. The integration with environments further extends these possibilities. When used as an initial state, pre-existing features of an environment can be exaggerated using contextually-based development. When used as an external environment, growths can adapt to the surrounding geometry, which in turn determines both small and large developmental features. Finally, the development of simulations within an environment can allow for the development of recognizable forms.

There are many possible extensions. To avoid the often exponential population growth present in the growths above, one could collapse edges to decrease the growth size. This would allow growths to achieve many more types of movement as if to emulate pulsing or breathing. Because the population wouldn’t be monotonically increasing, extended animated sequences would be possible.

It would also be interesting to investigate texture synthesis over the surface of the growth, allowing for increased realism of the output. When the start state is a textured mesh, there is a significant amount of color information that is discarded using the presented method. It would be fruitful to examine how color information could texture the growth during development, perhaps using a method similar to inheritance as described in Section 5.2.4. Another way to increase the visual complexity would be to layer runs of the growth on a variety of scales. For example, an initial growth could be run for a certain number of frames, after which the mesh could be subdivided and used as the initial state for another growth with different parameters. Other possibilities for expansion include particle orientation (a particle could exert a collision force dependent on its normal), more complicated topologies (tetrahedrons instead of triangles), or physically-based nutrient distribution methods (perhaps using fluid dynamics).
(a) Failed containment.

(b) Successful containment using exterior freezing.

Figure 23: Age-based reproduction.
A. Source Material

The child [3] and wolf [23] models were built by artists with usernames cvbtruong and WindTrees respectively, distributed by the 3D asset website TurboSquid [20]. Both models were posted under a royalty free license at no cost. For various applications in the presented method these two models were subdivided and otherwise manipulated.

The rock environment in Figure 21 was created by the author in January of 2017 in Mosaic Canyon, Death Valley National Park. At the time the author was supported by a Flint Initiative Grant through the Oberlin Conservatory.

B. Parameters

Table 1: Simulation parameters for presented figures. Note that the population refers to the number of particles, including environmental points.

<table>
<thead>
<tr>
<th>Fig.</th>
<th>Timestep</th>
<th>Pop</th>
<th>l</th>
<th>R</th>
<th>T</th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>c4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 (Left)</td>
<td>531</td>
<td>227669</td>
<td>0.4</td>
<td>2</td>
<td>20</td>
<td>0.1</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>5 (Right)</td>
<td>447</td>
<td>249992</td>
<td>0.4</td>
<td>2</td>
<td>20</td>
<td>0.1</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>6a</td>
<td>58</td>
<td>800</td>
<td>0.4</td>
<td>2</td>
<td>1</td>
<td>0.1</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>6b</td>
<td>220</td>
<td>61007</td>
<td>0.4</td>
<td>2</td>
<td>1</td>
<td>0.1</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>7c</td>
<td>339</td>
<td>445678</td>
<td>0.4</td>
<td>2</td>
<td>1</td>
<td>0.1</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>8a</td>
<td>18</td>
<td>18614</td>
<td>0.4</td>
<td>2</td>
<td>1</td>
<td>0.1</td>
<td>0.7</td>
<td>0.05</td>
<td>0.8</td>
</tr>
<tr>
<td>8b</td>
<td>33</td>
<td>76471</td>
<td>0.4</td>
<td>2</td>
<td>1</td>
<td>0.1</td>
<td>0.7</td>
<td>0.05</td>
<td>0.8</td>
</tr>
<tr>
<td>8c</td>
<td>74</td>
<td>809890</td>
<td>0.4</td>
<td>2</td>
<td>1</td>
<td>0.1</td>
<td>0.7</td>
<td>0.05</td>
<td>0.8</td>
</tr>
<tr>
<td>9</td>
<td>126</td>
<td>314812</td>
<td>0.4</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>10 (left)</td>
<td>0</td>
<td>25006</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>10 (center-left)</td>
<td>60</td>
<td>139199</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>10 (center-right)</td>
<td>105</td>
<td>667006</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>10 (right)</td>
<td>150</td>
<td>2970139</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>11a</td>
<td>120</td>
<td>1664213</td>
<td>0.5</td>
<td>1</td>
<td>800</td>
<td>0.1</td>
<td>0.9</td>
<td>0.01</td>
<td>0.3</td>
</tr>
<tr>
<td>11b</td>
<td>330</td>
<td>309127</td>
<td>0.5</td>
<td>1</td>
<td>1000</td>
<td>0.1</td>
<td>0.9</td>
<td>0.01</td>
<td>0.7</td>
</tr>
<tr>
<td>11c</td>
<td>330</td>
<td>120754</td>
<td>0.5</td>
<td>1</td>
<td>1000</td>
<td>0.1</td>
<td>0.9</td>
<td>0.01</td>
<td>0.7</td>
</tr>
<tr>
<td>12a</td>
<td>2100</td>
<td>136504</td>
<td>1</td>
<td>2</td>
<td>2000</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>12b</td>
<td>825</td>
<td>184587</td>
<td>1</td>
<td>2</td>
<td>2000</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>13</td>
<td>315</td>
<td>416024</td>
<td>0.4</td>
<td>2</td>
<td>10</td>
<td>0.1</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>14 (left)</td>
<td>210</td>
<td>33781</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>14 (center)</td>
<td>420</td>
<td>94287</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>14 (right)</td>
<td>735</td>
<td>1063874</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>15 (left)</td>
<td>60</td>
<td>26660</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>15 (center)</td>
<td>135</td>
<td>38892</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>15 (right)</td>
<td>225</td>
<td>285475</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>16a</td>
<td>3570</td>
<td>4811</td>
<td>0.4</td>
<td>1</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>16b</td>
<td>3570</td>
<td>7093</td>
<td>0.4</td>
<td>1</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Fig.</th>
<th>Timestep</th>
<th>Pop</th>
<th>$l$</th>
<th>$R$</th>
<th>$T$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16c</td>
<td>3570</td>
<td>20690</td>
<td>0.4</td>
<td>1</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>16d</td>
<td>3570</td>
<td>5461</td>
<td>0.4</td>
<td>1</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>16c</td>
<td>3570</td>
<td>14832</td>
<td>0.4</td>
<td>1</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>16f</td>
<td>3570</td>
<td>43029</td>
<td>0.4</td>
<td>1</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>17a</td>
<td>3315</td>
<td>1330</td>
<td>0.4</td>
<td>1</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>17b</td>
<td>13380</td>
<td>135154</td>
<td>0.4</td>
<td>1</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>18a</td>
<td>75</td>
<td>1662824</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
<td>0.99</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>18b</td>
<td>60</td>
<td>135706</td>
<td>0.4</td>
<td>2</td>
<td>1</td>
<td>0.1</td>
<td>0.7</td>
<td>0.05</td>
<td>0.8</td>
</tr>
<tr>
<td>19a</td>
<td>274</td>
<td>325057</td>
<td>1</td>
<td>2</td>
<td>20</td>
<td>0.105</td>
<td>0.95</td>
<td>0.05</td>
<td>0.8</td>
</tr>
<tr>
<td>19b</td>
<td>260</td>
<td>162321</td>
<td>1</td>
<td>1.6</td>
<td>20</td>
<td>0.105</td>
<td>0.99</td>
<td>0.05</td>
<td>0.8</td>
</tr>
<tr>
<td>20</td>
<td>45</td>
<td>31451</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>20 (left)</td>
<td>90</td>
<td>57146</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>20 (center-left)</td>
<td>150</td>
<td>178760</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>20 (center-right)</td>
<td>195</td>
<td>435990</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>20 (right)</td>
<td>240</td>
<td>1068286</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>21a</td>
<td>240</td>
<td>999236</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>21c</td>
<td>240</td>
<td>1039179</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>22a</td>
<td>375</td>
<td>1176303</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>22b</td>
<td>660</td>
<td>408497</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>22c</td>
<td>1020</td>
<td>1364943</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>23a (left)</td>
<td>210</td>
<td>695923</td>
<td>0.4</td>
<td>1</td>
<td>3</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>23a (middle)</td>
<td>465</td>
<td>1365074</td>
<td>0.4</td>
<td>1</td>
<td>3</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>23a (right)</td>
<td>675</td>
<td>3799870</td>
<td>0.4</td>
<td>1</td>
<td>3</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
<tr>
<td>23b</td>
<td>780</td>
<td>1470327</td>
<td>0.4</td>
<td>1</td>
<td>3</td>
<td>0.1</td>
<td>0.99</td>
<td>0.02</td>
<td>0.6</td>
</tr>
</tbody>
</table>
References


