Cellular Forms: an Artistic Exploration of Morphogenesis

Andy Lomas[1]

Abstract. Cellular Forms uses a simplified model of cellular growth to generate intricate sculptural shapes. Structures are created out of interconnected cells, with rules for the forces between cells, as well as rules for how cells accumulate internal nutrients. When the nutrient level in a cell exceeds a given threshold the cell splits into two, with both the parent and daughter cells reconnecting to their immediate neighbours. Many different complex organic structures are seen to arise from subtle variations of these rules, creating forms with strong reminiscences of plants, corals, internal organs and micro-organisms.

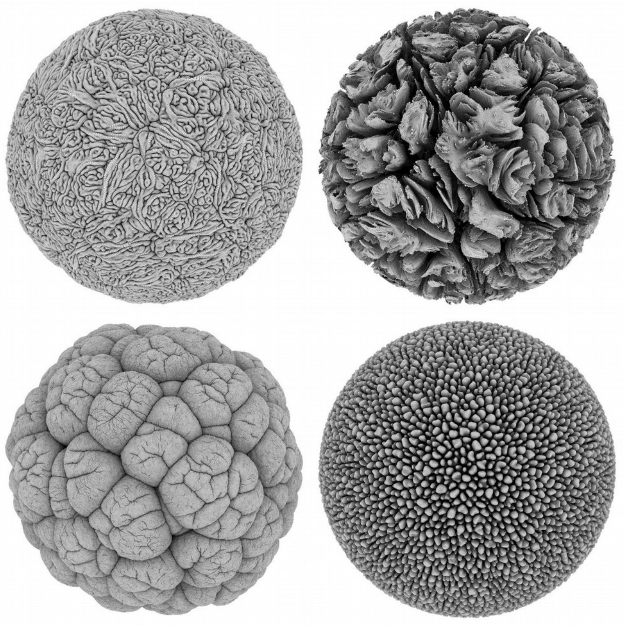


Figure 1. Examples of Cellular Forms.

The aim is to create structures emergently: exploring generic similarities between many different forms in nature rather than recreating any particular organism, and in the process exploring universal archetypal forms that can come from growth processes rather than top-down externally engineered design.

1 INTRODUCTION

The use of simulation methods in generative art can be seen as a natural extension of systems art [1], where the artist defines a process that can be run autonomously to create artefacts. With a sufficiently rich simulation system there is an expectation that surprising emergent results can be generated which would be difficult, or potentially even impossible, to create without the aid of digital technology. The process can be seen as one of exploration: both in defining the rules for the simulation systems and exploring the range of results that can be achieved once a system has been created.

In particular, simulation systems that are inspired by biological processes, such as morphogenesis, can be used as a powerful means to explore the nature of organic form. Can the astonishingly complex forms that are seen in nature emerge from simple rules? This can be viewed as exploring the nature of the basic fabric available to create structures when they are generated as a result of growth processes.

This paper describes Cellular Forms, an exploration of how rich evocative forms can be created using a simplified biological model of cellular growth. The model used is a deliberately simplified one, both to explore how a simple model can create richly emergent results, and to be computationally sufficiently efficient to allow the creation of structures with many millions of cells in order to achieve a high level of complexity and detail with the aim of evoking a powerful aesthetic result.

2 RELATED WORK

The relationship between growth and form has been the subject of study for many years. Major influences behind the work described here are Ernst Haeckel's studies of forms in nature [2] and D'Arcy Thompson's seminal “On Growth and Form” [3].

Alan Turing's paper “The Chemical Basis of Morphogenesis” [4] can be seen as the origin of using digital simulation methods to examine potential mechanisms behind pattern generation and growth, performing biological experiments “in silico” rather than “in vitro” or “in vivo”. The reaction-diffusion equations that Turing describes in his paper can create a surprisingly rich range of complex patterns, with remarkable resemblance to many of those seen in nature such as pigmentation patterns on the coats of leopards, zebra and angelfish. Variations on reaction-diffusion equations, particularly following the work of Greg Turk [5], are commonly used in computer graphics to create convincingly biological textures for creatures.

Probably the most common method utilised to create convincing biological structures is the use of L-Systems, as originally proposed by Aristid Lindenmayer [6]. Rules for branching and the relative sizes of segments between branches are expressed using a simple grammar with a generation rule that can be recursively applied to the form to create the next level. L-Systems are commonly used in computer graphics to produce tree-like forms, and can be a very efficient way to create complex structures, but effects like branching are explicitly

[1] Website: www.andylomas.com, Email: andylomas@yahoo.com

encoded into the system rather than arising emergently from lower level processes.

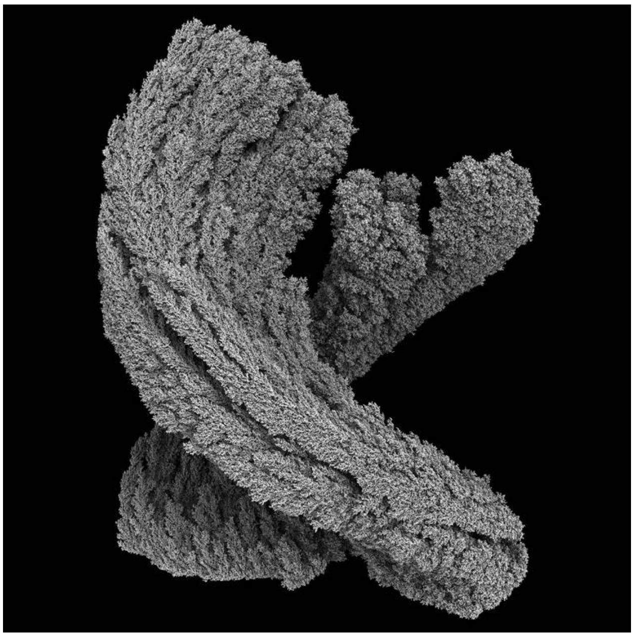


Figure 2. Example image from the Aggregation series.

Previous work by the author explored structures that can be created by variations on diffusion-limited aggregation [7], used as a simplified model of a growth system [8], [9]. This represents growth by repeated deposition: the structure is initialised with a single seed particle. Successive new particles are allowed to randomly move in an external medium until they collide with the structure generated so far. They then become attached to the structure at that position, and the process is repeated. The results of this Aggregation series were a range of structures with reminiscences of dendritic plants and finely branched corals.

The previous works most related to the work described here are Jaap Kaandorp's work on "Accretive Growth" [10] [11] and George Hart's "Growth Forms" [12]. Both of these use a model where cells are described by a surface of linked particles, with rules for when these cells split and how the topology of the surface changes after cell division. In Kaandorp's work the aim is to mimic the growth of coral-like forms, with growth in different areas based on external nutrient gradients that mimic the availability of water-borne food for marine organisms. In Hart's work some particles are designated as 'buds' which are used to explicitly control and stimulate growth in local areas, causing effects such as branching. The results of Kaandorp and Haart's work are a variety of branched coral-like structures, and the simulations described are run for a few thousand cell primitives.

3 CELLULAR FORMS

In Cellular Forms the principal aim was to create a system capable of generating complex biologically evocative forms based on the simulation of growth by cellular division.

Following the author's previous work, it was desired that the model should be flexible enough that it should be capable of producing results similar to those from the 'Aggregation' series as well as creating additional structures not achievable by that simulation framework. The aim was to be exploratory rather than create any specific target forms, but to be capable of producing structures reminiscent of internal organs such as the folded surface of brains.

In order to create this greater range of possible forms it was decided to use a model based on a simplified version of morphogenesis through cellular division. The system should be capable of creating complex sheets of cells, with rules governing when cells divide, how the topology of the surface of cells is affected by the newly created cells, and creating forces between cells to induce the surface to fold into complex shapes.

By having a variety of different methods to induce growth it was hoped to be able to create the extended range of structures desired. If growth was stimulated by external randomly transported food particles which directly cause the first cell they hit to divide, it should be capable of creating similar effects to those seen in the Aggregation series. On the other hand, if growth was based on concentrations of chemicals diffusing through the structure, potentially with all cells receiving the same amount of nutrient, then it was hoped that structures more like internal body organs could be produced.

It was also important that the simulation system should be capable of generating many millions of cell primitives. This was desired in order to achieve a compelling level of intricate detail in the structures, and had to be achievable within the limitations of available conventional PC hardware.

4 MATHEMATICAL MODEL

The model used is based on a simplified version of cells. A particle system [13] representation is used, with each cell represented by one particle, and each particle linked to a number of other particles that it is directly attached to.

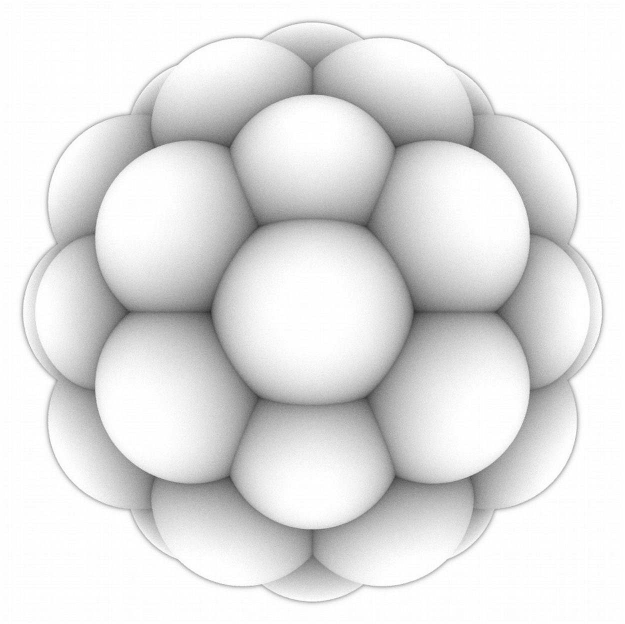


Figure 3. Initial ball of cells, with particles distributed uniformly on the surface of a sphere.

While the structures are three-dimensional, the topology of the connected particles is that of a two-dimensional surface. In all the structures illustrated here, the system starts with a simple ball of cells with all the cells uniformly spaced on the surface of a sphere.

Development of the form proceeds by a combination of forces that mediate interactions between the cells, and cell divisions that change the topology of the structure. The simulation takes place over time, with time incrementing in uniform clock cycle steps.

Cells that are directly linked try to maintain a constant distance from each other. Additional rules try to restore the sheet to a locally planar state if there is a fold in the surface, and to bulge the sheet out when links are in compression. The intention is that these two influences will work in competition with each other, with different strength factors for each tending to create surfaces with a variety of characteristics.

The actual implementation of all these effects is achieved by calculating a new target position for each of these influences, and offsetting the cell's position towards the new target position after multiplying by a restoring factor. Values for these factors are parameters for the simulation system as a whole.

Consider a system with simulation parameters *linkRestLength*, *springFactor*, *planarFactor* and *bulgeFactor.* Let a cell have position **P** which is linked to *n* particles with positions $\mathbf{L}_r$, and the unit length normal to the surface at the current cell position is **N**. The target positions for three different influencing effects are calculated by the following methods:

1. The tendency for linked cells to maintain a constant distance from each other is implemented using a linear spring-like system. The target position for the springs is calculated by taking the average of the rest positions that each link would push the particle to if it were the only influence:

$$\mathbf{springTarget} = 1/n \sum_{r=1}^{n} \left(\mathbf{L}_r + linkRestLength \times \widehat{(\mathbf{P}-\mathbf{L}_r)}\right)$$

2. The planar target position, that acts in a similar manner to a torsion spring, is simply calculated by taking the average of all the positions of directly linked particles. This is designed to have the effect of tending to reduce folds and bumps in the surface, restoring the surface to a local planar state:

$$\mathbf{planarTarget} = 1/n \sum_{r=1}^{n} \mathbf{L}_r$$

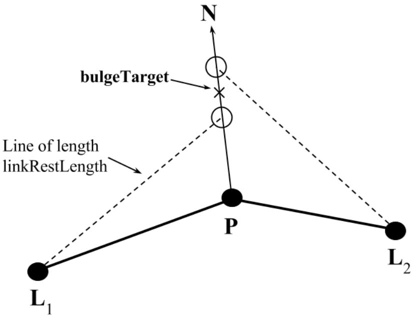


Figure 4. Diagram illustrating calculation of the bulge target position.

3. The bulge target position is determined by calculating the distance that each link would have to push the particle outwards along the direction of the surface normal in order to restore the link to its rest length. This is designed to have an effect of tending to bulge the surface outwards in the direction of the normal when links are in compression. The bulge distance due to each link is calculated by a simple application of the cosine formula for triangles, and the average taken for all the links to create the desired overall distance *bulgeDist* in the direction of the normal. The **bulgeTarget** vector is then taken by going *bulgeDist* in the direction of the surface normal from **P**.

$$dotN_r = (\mathbf{L}_r - \mathbf{P}) \cdot \mathbf{N}$$

$$bulgeDist = 1/n \sum_{r=1}^{n} \left(\sqrt{linkRestLength^2 - |\mathbf{L}_r|^2 + dotN_r^2} + dotN_r\right)$$

$$\mathbf{bulgeTarget} = \mathbf{P} + bulgeDist \times \mathbf{N}$$

The new position for the particle position **P'** is then calculated by offsetting **P** in the direction of each of these target position using the three different simulation factor values.

$$\begin{aligned}\mathbf{P}' = \mathbf{P} &+ springFactor \times (\mathbf{springTarget} - \mathbf{P}) \\ &+ planarFactor \times (\mathbf{planarTarget} - \mathbf{P}) \\ &+ bulgeFactor \times (\mathbf{bulgeTarget} - \mathbf{P})\end{aligned}$$

One thing to notice about this is that there is no momentum term: the new position is simply calculated by moving it a fraction of the distance towards the target positions controlled by the restoring factors. This can be justified if we consider the cells to be growing in a medium that is highly viscous relative to the cell size, so there is a large damping effect on any velocities. It also has the advantage of making the system less prone to unstable oscillations, particularly when we are in effect adding energy to the system every time cells divide.

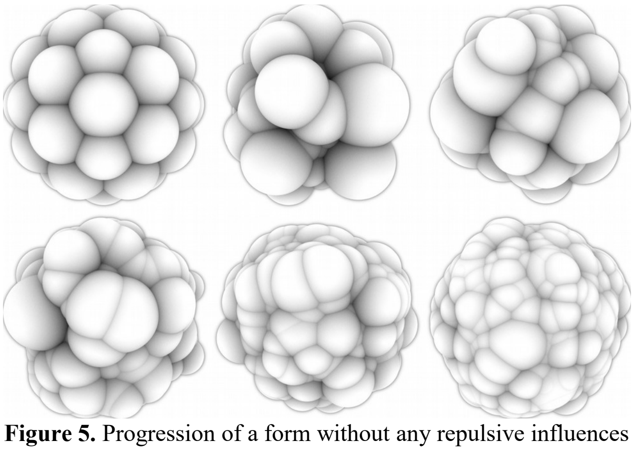


Figure 5. Progression of a form without any repulsive influences between close cells. Structure degenerates into incoherent state.

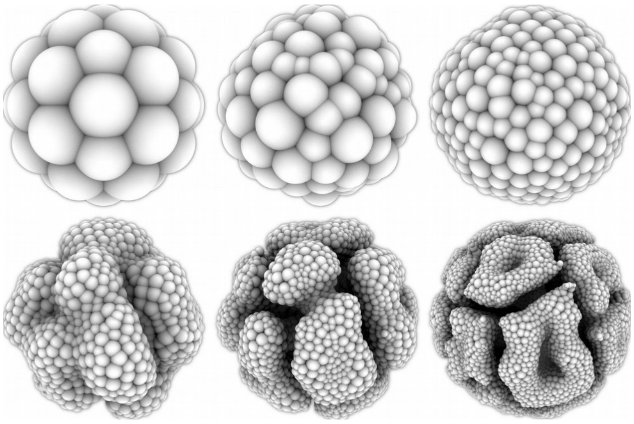


Figure 6. Progression of a form with the same parameters but with repulsive influence between close cells. Structure develops coherent shape.

Cells that aren't directly linked to each other but are in close proximity experience a repulsive influence. Without this it was found that structures would degenerate into an incoherent state as they started to generate folds. In effect the repulsive influence between cells imposes a constraint of structural coherence on the form. An analogy can be drawn with D'Arcy Thompson's arguments for how physics imposes constraints on the possible shapes for forms created by growth processes.

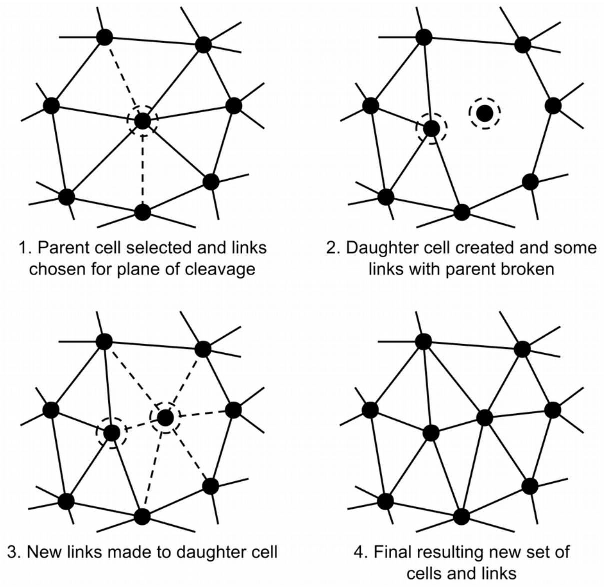
This repulsive influence is controlled by two other parameters for the simulation system which define a radius of influence (*roi)* and *repulsionStrength*. The effect of the repulsion is applied by calculating a **collisionOffset** vector to be added to each particle's position where

$$\mathbf{collisionOffset} = repulsionStrength \sum_{r \in A} \left(\frac{roi^2 - |\mathbf{P} - \mathbf{P_r}|^2}{roi^2} \times \widehat{(\mathbf{P} - \mathbf{P_r})} \right)$$

and *A* is the set of all particles within the radius of influence of the current particle that aren't directly linked to the current particle. Directly linked particles are excluded from the repulsion calculations since they are considered directly attached to each other, and the influences between them are already controlled by the previously described other effects.

Each cell has an internal 'food level'. The food accumulates in a cell and when it exceeds a given threshold the cell is selected for splitting. Various different methods have been implemented to affect food levels, and therefore affect growth rates, including:

- Uniformly adding a random amount to each cell at each time step.
- Using reaction-diffusion equations (RDEs) over the surface [5] to create differential areas of growth. The RDE is calculated using the cells as the places where chemicals are stored, and diffusing the chemicals along the direct links between cells. One of the RDE chemicals is used as a 'nutrient level', affecting how much the food level in the cells increments at each time step.
- Using ray-tracing to simulate light coming in from outside the structure which stimulates nutrient creation in the cells that the light hits. This nutrient is used to control the food increment in each cell, and can also diffuse from one cell to another along the links between cells to distribute the nutrient through the structure.



1. Parent cell selected and links chosen for plane of cleavage

2. Daughter cell created and some links with parent broken

3. New links made to daughter cell

4. Final resulting new set of cells and links

Figure 7. The process of cell division.

When a cell has been selected for splitting, a number of parameters are used to control how the split occurs. These are used to control how the topology of links between the cells is changed by cell division. First, two links are chosen that represent the plane of cleavage. All the links to one side of the plane of cleavage are left connected to the parent cell, while the links to the other side are disconnected from the parent and replaced with links to the daughter cell. Along the plane of cleavage links are made to both the parent and daughter cells. A new link is also created directly between the parent and daughter.

There are some key simplifications in this system compared with one that aims to accurately mimic biological processes. In particular:

- Apart from the potential for differential growth rates as described above, there is no cell differentiation. The parameters governing forces between cells and how cell division takes place are constants for the whole system, and don't change over time. The target link length is also the same uniform constant value for all links between any cells.
- Cells are represented by simple spheres with only a position and radius. They don't have any principal axis or orientation.
- There is no cell motility. The links that describe attachments between cells are formed immediately following cell division.
- As previously mentioned, the cells are always connected together in arrangements that are topologically equivalent to a simple closed surface rather than being a solid volume.

5 IMPLEMENTATION

Typical simulations are run for tens of thousands of iterations with data sets growing to over 50 million particles.

The software to run the simulations and render images from the data sets created is implemented in C++ and CUDA [14].

To make use of the general purpose parallel processing capabilities of modern graphics hardware, all the calculations involving the individual particles are executed using CUDA on the GPU. This includes all the functions that simulate the effects of forces between the particles, cast rays into the structure to render images or simulate light rays, and handle topological changes to the surface that occur when cells divide. This means that all the data for the cellular structures can be kept purely in GPU memory, avoiding the need to transfer large amounts of data across the PCI bus between the host (CPU) memory and the device (GPU) memory. In the current implementation the only need for transferring cell data between the CPU and GPU is if it is required to write out data to a file on disk, or to read data back from disk.

In particular, implementing using the GPU allowed a very significant speed improvement for the calculations of repulsive interactions between cells in close proximity but not directly linked to each other. The previous implementation of this on the CPU was too slow to make it feasible to run simulations with the required number of cells.

Class name	**Class description**
BaseParticleGpu	Base class defining particle data to be held on the GPU.
LinkedParticleGpu	Derived from BaseParticleGpu. Extends the base class by adding a list of links between particles. Also implements support for reaction-diffusion equations using the links between particles.
ElasticSheetParticleGpu	Derived from LinkedParticleGpu. Adds methods for dealing with the spring, planar and bulge effects, as well as repulsion effects between spatially close particles.
FoodSplitParticleGpu	Derived from ElasticSheetParticleGPU. Adds methods to implement food levels, nutrient levels, and cell division.

Table 1. Class structure used for particle data.

The software is implemented as an extensible framework. An object oriented approach was used with a hierarchy of inheriting classes representing the data for the structures. This starts with a base class for representing general particle data on the GPU, which is then specialised through a series of derived classes that inherit from each other up until the final class used in all the simulations shown here.

Rendering 2D images from the cellular simulation data is done using ray-tracing techniques, with the cells treated as spheres with radii based on the average distances to linked cells. A number of different techniques are implemented, including producing solid surface renders with 'ambient occlusion' for the shading which represents the effects of a self-shadowing from a uniform omnidirectional diffuse light, and “X-Ray” renders that treat each sphere as a contributor to an accumulated density calculated by tracing a ray though the whole structure. All the key rendering functions are implemented using CUDA kernels on the GPU, both for speed and so that all the particle data can be kept solely on the GPU.

6 ARTISTIC ARTEFACTS

Simulations are first run with a maximum of a million cells to create quick initial sample tests of simulation parameter values which may create interesting results. These tests typically take between 1 and 5 minutes to run for each sample. From these, candidates are selected to run full length simulations and high-resolution renders. These final simulations are executed to create the most detailed structures possible, which typically means between 52 million and 56 million cells before the memory limits of the current hardware (NVIDIA GTX Titan with 6GB RAM) are reached. These full length simulations typically take between 1 and 4 hours each.

Rendered images are created directly from the simulations, which are used to make the artistic artefacts from the series. These artefacts can take various forms:

- High resolution prints of the final structures. Currently these are rendered at 8600 by 8600 pixels, to allow very fine detail when printing at large sizes.
- High definition animations taken by rendering images at equally spaced time intervals, to show how the structures develop incrementally over time.
- 3D stereo views taken by rendering pairs of images using cameras with an interocular separation.

Currently, two main rendering styles are used for the final artefacts:

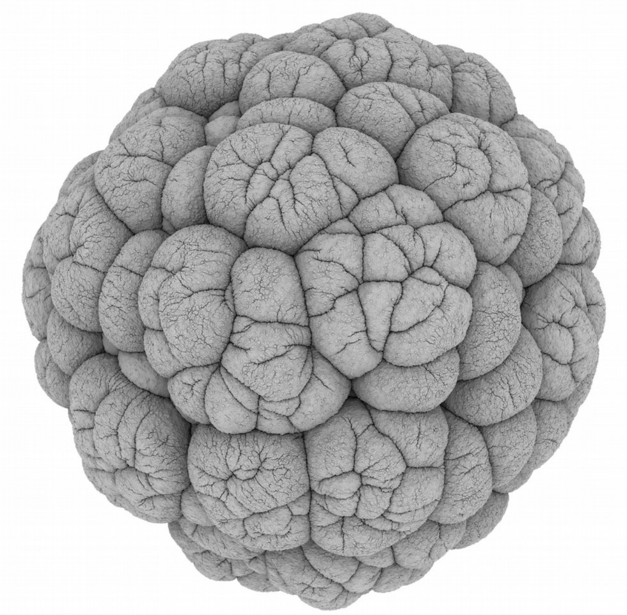


Figure 8. Cellular Form rendered using solid surface and ambient occlusion shading.

1. Solid surfaces that use 'ambient occlusion' self-shadowing to reveal the final forms. Uniform omni-directional diffuse light is used on a surface with Lambertian (ideal diffuse) reflectance to create a directionally unbiased view of the surface. The images produced in this manner emphasise sculptural forms of the surfaces, and echo the illustrative techniques that Ernst Haeckel used in his studies of nature [2].

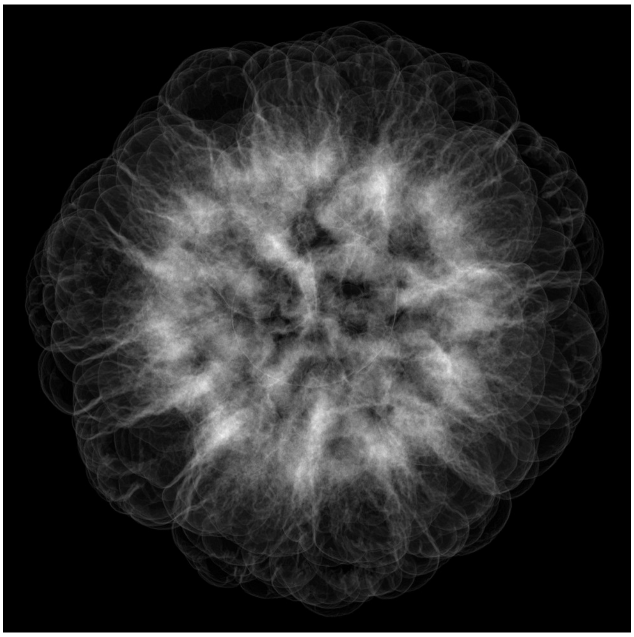


Figure 9. The same simulation data as Figure 8 rendered using an X-Ray style density accumulation map.

2. 'X-Ray' images that show accumulated density through the whole structures. These images reveal complex internal structures not apparent in the external solid renders, and have strong resemblances to biological studies using microscopy.

7 RESULTS

The structures produced by the simulations are all simple closed surfaces, topologically equivalent to the surface of a sphere. However, they become incredibly intricately folded due to tensions in the surface and forces between the cells as the structure develops over time.

The original intention behind creating this system was to be able to create a greater range of emergent forms than had been seen with previous work that explored variations of diffusion-limited aggregation as a simplified model of growth systems [8], [9].

The results have exceeded the initial expectations. Many rich behaviours appear to emerge such as complex folding, branching, rhythmic pulses and waves of growth propagating across surfaces. These can be particularly observed in the animations created from the simulations [15], [16]. Many of these effects appear to have deep resemblances to behaviour seen in nature such as pulses of growth during embryonic cell division [17] and mitotic division waves [18].

One thing to note is that all the structures seen here appear to emerge without the need for differentiation into different cell types. The system framework allows reaction-diffusion equations to be used to create regions of cells with different growth rates, but it was found that these influences could be completely omitted and still produce rich results. All the forms shown here were created without the use of RDEs, only using either uniform growth rates for all cells, or having nutrient for cell growth created by incident light rays which could diffuse throughout the structure.

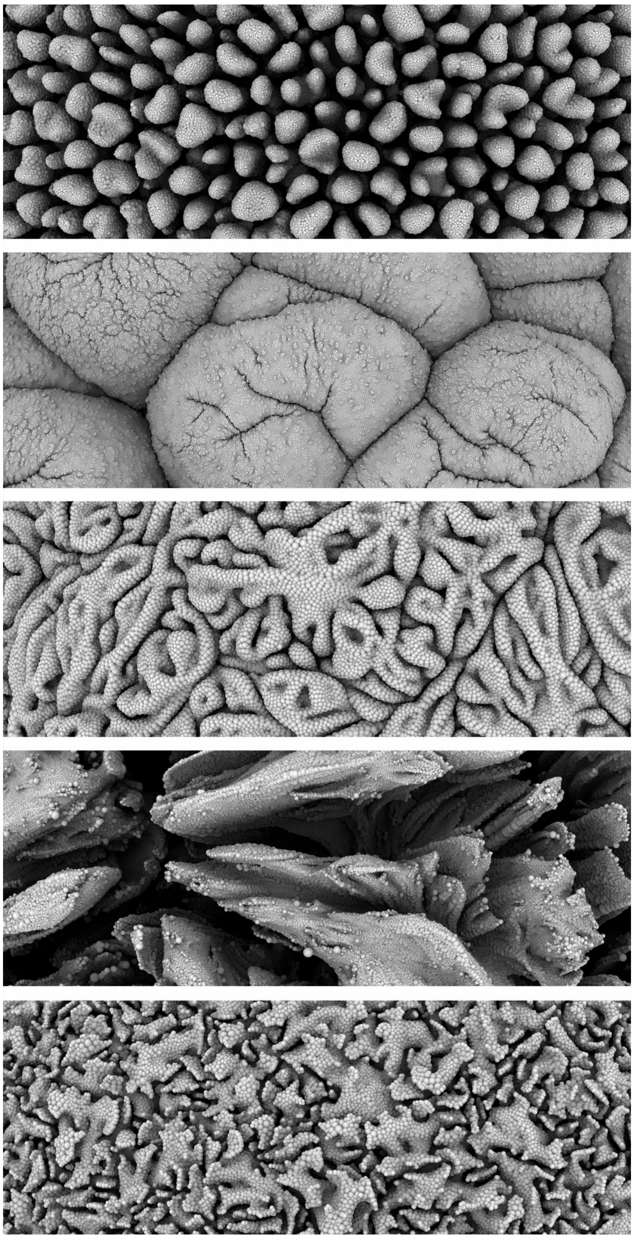


Figure 10. Examples of different surface patterns and shapes on Cellular Forms.

In general, using uniform cell growth rates appears to produce results that look more like internal organs, such as brains, with complex folded shapes. Using incident light rays to create nutrient produces a larger range of structures. This includes the internal organ-like structures when the inter-cellular diffusion rate for nutrient diffusion rates are high, but when it is set low and the light comes from a single direction we appear to naturally get remarkably plant-like forms.

One of the most necessary contributions to create interesting structurally coherent forms appears to be the use of repulsion influences between cells which are in close proximity but not directly linked to each other. Without these the forms tend to degenerate into a meaningless unstructured mess, as there is nothing preventing the surface self-intersecting as it folds. These additional influences can be seen as exerting physical constraints on the structures.

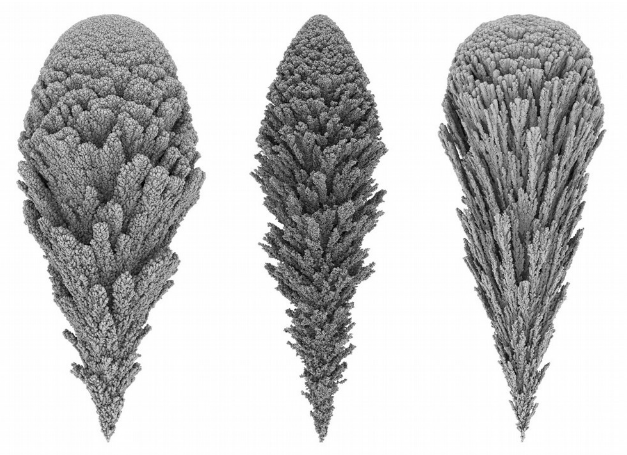


Figure 11. Examples of plant-like forms with nutrient created using light from a single direction.

8 CONCLUSIONS & FUTURE WORK

The use of a simple system of interactions between adjacent or spatially close cells is seen to produce a wide range of complex results.

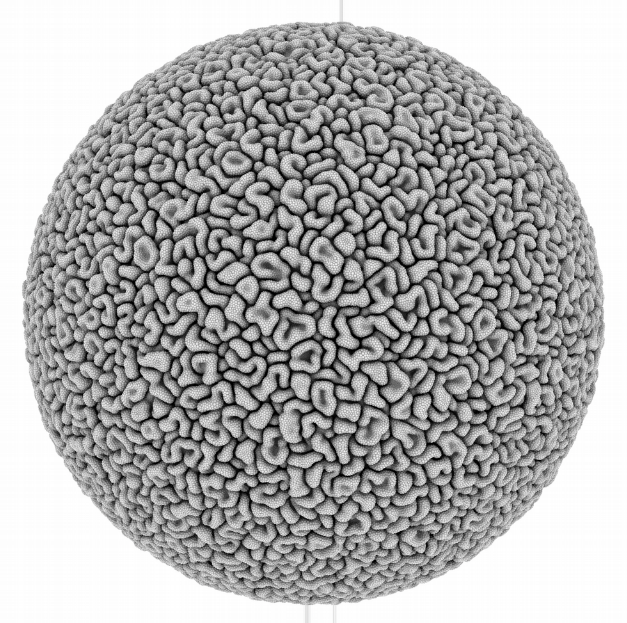


Figure 12. Cellular Form with surface shapes reminiscent of patterns formed by reaction-diffusion equations.

Many of the surface patterns produced, such as protuberances and brain-like folds, are reminiscent of the sort of shapes produced by reaction-diffusion equations. It is potentially interesting that these are seen to occur without the need for any explicit influencers reacting together and diffusing between the cells.

The framework used here is an extensible one and there are a number of interesting future directions that could be taken.

The current model doesn't have any active cell differentiation, with the simulation parameters set globally for the whole system. Cell differentiation could be explored in a number of different ways, such as by having different simulation parameters for each cell. One simple idea would be to have two sets of parameters, and a gradient between the cells which is used to blend between those parameter values for each cell. It may also be interesting to re-introduce reaction-diffusion equations to control the blending between the parameters instead of having a single simple gradient.

Additional or alternative implementations of the forces between cells could be explored. It may be reasonable to expect that many of the structures seen here are generic enough to not be the results of specific details of the implementations used, but it would be interesting to see if the types of structures generated do change if modifications are made to the inter-cellular forces.

Currently the main limit to the complexity of the structures created is the available memory in the GPU to hold the cell data. With increasing hardware capabilities it will be possible to run larger simulations. The algorithms used could also be changed to remove the requirement that all particles are held concurrently in GPU memory, though this may have impractical performance implications, or be altered to distribute calculations between multiple GPUs.

The effect of other influences and forces on the system could be explored. These could be readily integrated into the system using similar mechanisms to the current implementation of repulsion between cells. Interesting sources could be fluid flows, surfaces that exert their own attraction or repulsion forces, or the cells themselves acting as primitives that can generate their own complex electrical or magnetic force fields which affect subsequent growth.

It would be interesting to explore the effects of cell death as well as cell growth, with appropriate topological changes to the surface when cells die. These could include rules that allow holes to be created in the surface, resulting in the surface no longer being equivalent to the surface of a sphere.

The software framework could also be extended to model cells in volumetric arrangements, such as by using tetrahedral meshes. However, since the prime goal here was to create interesting sculptural shapes, the use of a surface model was probably appropriate. This can also be justified by taking into consideration how epithelial cell tissues in real biological systems undergo morphogenesis.

Other rendering techniques could be used. One alternative that was explored in early tests was to use a variation on Voronoi cells [19] to illustrate the cell walls [20]. This isn't currently implemented on the GPU, but it should be a relatively easy addition to the code used for rendering.

It would be fascinating to see if 3D printing could be used to turn these structures into physical sculptural form, revealing their full three-dimensional complexity. Some of the more detailed dendritic forms are likely to prove challenging, but it should be an interesting challenge.

REFERENCES

[1] Wikipedia article on Systems Art: http://en.wikipedia.org/wiki/Systems_art

[2] E. Haeckel, *Art Forms in Nature*, Dover Publications Inc.; Revised edition (2 Jan 2000).

[3] D.W. Thompson, *On Growth and Form*, Cambridge University Press; New Ed edition (31 July 1992).

[4] A.M. Turing, *"The Chemical Basis of Morphogenesis"*, Philosophical Transactions of the Royal Society of London 237 (641), 37–72.

[5] G. Turk, *"Generating textures on arbitrary surfaces using reaction-diffusion"*, Proceedings of SIGGRAPH '91, 289-298.

[6] A. Lindenmayer, *"Developmental systems without cellular interaction, their languages and grammars".* Journal of Theoretical Biology, 30 (1971), 455-484.

[7] T.A. Witten Jr & L.M. Sander, *"Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon"*, Phys. Rev. Lett. 47, 1400 (1981).

[8] A. Lomas, *"Aggregation: Complexity out of Simplicity"*, sketch session SIGGRAPH 2005, http://www.andylomas.com/sketch_0087_final.pdf

[9] A. Lomas, video: *"Growth by Aggregation"*, https://vimeo.com/83297099

[10] J.A. Kaandorp, *"Analysis and synthesis of radiative accretive growth in three dimensions",* Journal of Theoretical Biology, 175 (1995), 39-55.

[11] J.A. Kaandorp & J. Kübler, *The Algorithmic Beauty of Seaweeds, Sponges, and Corals*, Springer-Verlag (2001), 125-144.

[12] G. Hart, *"Growth Forms"*, Proceedings of Bridges 2009, http://georgehart.com/Growth/GrowthForms.pdf

[13] W.T. Reeves, *"Particle Systems: A Technique for Modelling a Class of Fuzzy Object"*, ACM Transactions on Graphics, Volume 2 Issue 2, April 1983, 91-108.

[14] CUDA http://www.nvidia.com/object/cuda_home_new.html

[15] A. Lomas, video: *"Cellular Forms"*, https://vimeo.com/82989945

[16] A. Lomas, video: *"Cellular Forms (X_Ray version)"*, https://vimeo.com/83294152

[17] Video: *"The Development of a Frog"*, http://www.youtube.com/watch?v=dXpAbezdOho

[18] Video: *"Early mitotic division waves in Drosophila embryo",* http://www.youtube.com/watch?v=-Nf6CyWNodA

[19] Wikipedia article on Voronoi diagrams: http://en.wikipedia.org/wiki/Voronoi_diagram

[20] A. Lomas, video: *"Cellular Form Generation Test 003c5"*, http://www.youtube.com/watch?v=kcEBEIAhsuk