An Emergent System for Self-Aligning and Self-Organizing Shape Primitives

Linge Bai
Manolya Eyiyurekli
David E. Breen
Department of Computer Science
Drexel University
3141 Chestnut Street, Philadelphia, USA
{lb353, me52, david}@cs.drexel.edu

Abstract

Motivated by the natural phenomenon of living cells self-organizing into specific shapes and structures, we present an emergent system that utilizes evolutionary computing methods for designing and simulating self-aligning and self-organizing shape primitives. Given the complexity of the emergent behavior, genetic programming is employed to control the evolution of our emergent system. The system has two levels of description. At the macroscopic level, a user-specified, pre-defined shape is given as input to the system. The system outputs local interaction rules that direct morphogenetic primitives (MP) to aggregate into the shape. At the microscopic level, MPs follow interaction rules based only on local interactions. All MPs are identical and do not know the final shape to be formed. The aggregate is then evaluated at the macroscopic level for its similarity to the user-defined shape. In this paper, we present (1) an emergent system that discovers local interaction rules that direct morphogenetic primitives (MP) to aggregate into the shape, (2) the simulation system that implements these rules and causes MPs to self-align and self-organize into a user-defined shape, and (3) the robustness and scalability qualities of the overall approach.

1. Introduction

In developmental biology, living cells self-organize and form into specific shapes and structures. Cells may emit a chemical that diffuses into the surrounding environment. Neighboring cells detect the chemical concentration at their surfaces and respond to the chemical stimulus. This phenomenon is known as chemotaxis. Another phenomenon observed in cell aggregation is cell alignment, i.e. cells tend to align with each other in order to create local patterns and structures. In this self-organizing emergent behavior, a large number of identical cells interact with each other locally without the use of external or global information, such as a description of the macroscopic structure being formed or the final position of each cell within the structure. Motivated by the natural phenomenon of self-organizing cell-level emergent behavior, we present an emergent system that allows morphogenetic primitives (a type of virtual cell/agent) to self-align and self-organize into a user-defined shape.

Our system explores two levels of emergence: pattern formation and intrinsic emergence [7]. Pattern formation is an observer-based self-organization process. An external observer identifies and evaluates unexpected patterns generated by the system and the system itself does not necessarily benefit from these macroscopic patterns. Intrinsic emergence refers to the emergent behaviors that are useful within the system. In an intrinsic emergent process, local interactions support a global information process, which in turn benefits individuals in the system. For example, the coordinated behavior of a school of fish allows global information processing through local interaction, even though there is no leader in the system. In our system, we extend pattern formation into shape composition [2]. Given a specific shape, MPs in the system self-organize into an aggregate with a similar shape using only local interactions. In terms of intrinsic emergence, given random initial orientation, MPs are
able to self-align and aggregate into a shape with a global coordinate system.

Inspired by cell biology, we have developed an emergent system for self-aligning and self-organizing shape primitives. The interactions of the virtual cells/agents, called Morphogenetic Primitives (MPs), are based on chemotaxis-driven aggregation behaviors exhibited by actual living cells. Cells emit a chemical into their environment. Each cell responds to the stimulus by moving in the direction of the gradient of the cumulative chemical field detected at its surface. MPs, though, do not attempt to completely mimic the behavior of real cells. The chemical fields are explicitly defined as mathematical functions and are not necessarily physically accurate.

In our system, MPs have their own coordinate system and are at first randomly placed in a 2D toroidal environment with random orientation. All MPs are logically identical in that they have the same prescribed behavior and are equally important in the emergent process. There is no “seed” MP and there is no special region/information in the toroidal environment. Each MP is an independent entity that senses its local chemical field, responds to it, and then modifies the environment and its internal state. Each MP also aligns itself with its immediate neighbors. An MP identifies its neighbors’ orientations, calculates the average orientation of these neighbors and incrementally rotates itself towards the average orientation. Figure 1 presents an example of a macroscopic shape and orientations that has emerged from local MP interactions. A collection of MPs are initially randomly positioned and oriented. The orientation of the primitives is visualized with color.

Given the complexity of the MPs’ self-organizing behavior, we employ an evolutionary computing method [11] to design the emergent system. The first level of emergence of the system, shape composition, is an observer-based approach. A user-defined shape is the input to the system. MPs then self-organize into an aggregate with a similar shape. The challenge is to discover the local interaction rules that direct MPs to self-organize. An evolutionary computing method is a good candidate to meet this challenge because selection may be used to direct the evolution. The individuals evolved in the system are the local interaction rules that direct MPs to self-organize. The positive and negative feedback cycles [19] in the self-organizing system can be represented as local interaction rules of high and low fitness values. The evolutionary system prefers individuals with higher fitness values over ones with lower fitness values, and given enough time and proper evolution parameters discovers a local interaction rule that directs MPs to aggregate into the pre-defined, macroscopic shape.

The design of the emergent system makes use of the positive and negative feedback cycles of self-organizing nonlinear systems [19]. To effectively direct and control the self-aligning and self-organizing system of shape primitives, genetic programming [21] is employed as an observer that distinguishes “good” and “bad” aggregates, i.e. it identifies the positive and negative feedback cycles. The system can be viewed as a framework for designing a self-organizing system for automated shape composition. It can also potentially be applied to the control of robot swarms and the computer-aided design of real cell aggregates.

2. Related Work

Camazine et al. provided a definition of self-organization for pattern formation in biological systems [6]. They pointed out that from numerous local interaction between low-level components of a biological system, patterns appear at the global level. Moreover, the interactions among these low-level components use only local information, without reference to the global pattern. Self-organization is accompanied by emergence of patterns and structures. Crutchfield [7] specified a distinction between two kinds of emergence: pattern formation and intrinsic emergence.

Several methods have also been proposed for designing self-organizing systems. Haken [18] developed the theory of synergetics, which describes pattern formation in complex systems. He introduced order parameters to explain self-organizing structures in nature. Crutchfield and Mitchell proposed using genetic algorithm to evolve optimal strategies [8]. They make use of GA’s ability to explore large search spaces and demonstrated that evolutionary design is capable of discovering order parameters that lead to macroscopic patterns using selection and genetic variation of microscopic components.

Theraulaz and Bonabeau presented a modeling approach based on the swarming behavior of social insects [23, 24], a type of swarm intelligence [3]. They combined swarm techniques with 3D cellular automata to create autonomous agents that indirectly interact in order to create complex 3D structures. This indirect interaction, known as stigmergy [25], allows the agents to act cooperatively, but independently, through a stimulus-response mechanism based on modifications made to the environment. Bonabeau et al. [4] applied genetic algorithms (GA) to the stigmergic swarm-based 3D construction method in order to improve the overall process. A fitness function, chosen by human observers, is assigned to each pattern in this approach, and GA is used to search the space of all possible patterns.

Nagpal et al. present techniques to achieve programmable self-assembly [22]. Cells are identical, programmed units which are randomly distributed and communicate with each other within a local area. In this approach, global to local compilation is used to generate the program executed by each cell, which has specialized, predetermined initial variables that reflect special regions of
the environment.

Fleischer explored a cell-based developmental model for self-organizing geometric structures [14, 15]. He applied his cell interaction simulation system to computer graphics to produce an approach to cellular texture generation [16]. Egenberger Hotz proposed the use of genetic regulatory networks coupled with developmental processes for use in artificial evolution and was able to evolve simple shapes [10, 20]. The combination of artificial evolutionary techniques and developmental processes provides a comprehensive framework for the analysis of evolutionary shape creation.

We have previously developed a computational model and software system that is capable of simulating chemotaxis-based cell aggregation and sorting in 2-D [12, 13]. Our 2-D model incorporated fundamental parameters involved in cell-cell aggregation, such as a cell’s ability to emit and detect chemoattractant chemicals, cell motility, attachment, proliferation, aggregation and various stages of a cell’s life cycle. This model and system provide the conceptual and software foundation for morphogenetic primitives. We modified and extended the cell simulation system to produce the self-aligning and self-organizing simulation system needed for MPs. The local interaction rules of MPs are explicitly defined as the chemical field around the primitives. The system discovers the local interaction rules that direct MPs to form into a pre-defined, macroscopic shape.

Our work is conceptually novel in that it extends pattern formation into shape composition. Instead of predicting complex patterns from emergent behavior, our system takes the final shape as an input and outputs the rules/methods that direct the primitives to self-organize into that shape. Unlike Bonabeau’s work which requires human input to perform fitness evaluation, our system automates this process by computationally comparing the similarity of the aggregate and the shape. Unlike Nagpal’s work that requires a special global initialization of the environment and cell states in order to differentiate the cells, no special regional information is required in our system and all primitives are exactly the same. Additionally our shape primitives do not contain or utilize a representation of the shape to be formed. Unlike Fleischer’s work which cannot automatically generate user-defined shapes, our system has successfully generated a number of pre-defined macroscopic shapes, i.e. an ellipse, a diamond, an hourglass and a boomerang, as well as unexpected shapes and patterns, using genetic programming to explore the space of chemical field functions that surround the individual primitives. These individual fields create a cumulative field that directs the motions of the MPs. The final user-defined shape emerges from the collective actions of the MPs.

3. System Description

The main challenge in our approach is how to design MPs such that they will perform local actions that produce a macroscopic shape without the cell/agents having any specific knowledge or representation of the shape being formed. We discover the local interaction rules that lead MPs to self-organize into an aggregate of the desired shape. Our approach gives rise to two forms of information processing in the system: top-down and bottom-up. In the top-down direction, the system transforms the concept of a macroscopic shape into the local interaction rules of MPs. In the bottom-up direction, MPs perform local actions which make them self-align and self-organize into an aggregate. The aggregate is then compared with the final shape at the macroscopic level in the genetic programming’s fitness evaluation process.

The two major components of our system are a genetic programming framework and a self-aligning, self-organizing simulation system. The local interaction rules that direct MPs to aggregate into shapes and patterns can be explicitly represented as a “chemical” field function. In the context of a GP framework, the individuals to be evolved are the “chemical” field functions. Initially, the GP framework randomly generates a population of functions and then parses each of them into a “chemical” field function. Each function is compiled into the self-aligning, self-organizing simulation program, defining the “chemical” field around individual primitives. After the primitives form into an aggregate, each aggregate is then compared with the pre-defined shape through the GP fitness evaluation. The similarity between each aggregate and the pre-defined, macroscopic shape is viewed as the fitness value of the individual, i.e. the field function. The GP framework then performs selection and variation based on the fitness values in order to define the next generation for the simulation computation. The system stops when the GP termination criteria is met, that is, either the system generates an aggregate similar to the pre-defined shape or a certain number of generations have been produced. This procedure is summarized in Algorithm 1.

4. Local Interactions Between MPs

4.1. Design Principles

Several principles were followed when developing morphogenetic primitives. 1) MPs are autonomous “agents”. Each MP is an independent entity that senses the environment, responds to it, and then modifies the environment and its internal state. There is no “master designer” directing the actions/motions of the MPs. 2) Actions are based on local information. Each primitive emits a finite field that can
Algorithm 1 System Framework

The macroscopic shape is given as input
Initialize individuals (chemical field functions)

while termination criteria not met do
    $i \leftarrow 0$
    while $i < \text{population size}$ do
        Compile individual $i$ into MP simulation system
        Run MP self-organizing simulation
        Compare MP aggregate with predefined shape
        Assign the similarity as fitness value
        $i \leftarrow i + 1$
    end while
    Record the best fit individual and MP aggregate
    Perform crossover and mutation
    Generate a new generation
end while
Output the individual with the best fitness value and its aggregate

be sensed only by other primitives within a certain range. The only information received by an MP is gathered at its surface, namely the concentration of the cumulative field and contact with immediate neighboring MPs. 3) MPs respond to information with prescribed behaviors. The actions performed by each MP are the same, but the specifics of the individual actions are based on information received from the environment. 4) MPs have no representation of the final, macroscopic shape to be produced. MPs do not use information about the final shape to determine what actions to take. Their actions are pre-determined by the ultimate shape to produce, but MPs do not carry or access information about the shape. The MP’s final global position relative to the shape is not known ahead of time. 5) The shape emerges from the aggregation of local interactions and behaviors. Rather than follow a plan to produce the shape, MPs sense, change and respond to the cumulative field concentration. This simple behavior, when combined with somewhat complex individual fields, will direct the MPs to take individual actions based on local information that will ultimately aggregate to produce a user-defined, macroscopic shape.

Each MP simulation process begins by randomly placing a number of MPs (500 for our examples) with random local orientation in the computational environment. A morphogenetic primitive is represented by a small disk existing in a toroidal 2D environment. The environment is effectively infinite with no boundaries, since the top edge of our computational world is connected to the bottom edge, and the left edge is connected to the right edge. A single aggregation simulation is comprised of a series of time steps. For each time step, each cell performs a prescribed set of actions.

4.2. Chemical Emission

Each MP emits a “chemical” into the environment. The “chemical” is secreted from the MP’s surface and diffuses within a certain distance of influence in all directions in 2D. In our system the field function is truncated at a fixed distance ($R_{\text{max}} = 200$ units) in order to keep the MP interaction finite and local. The “chemical” field emitted by one type of MP is presented in Figure 2. Here the darker portions of the field represent regions of higher chemical concentration.

The “chemical” field of the individual MPs accumulates in the environment. It is this cumulative field that each MP senses and responds to. Figure 3 illustrates the cumulative field produced by a collection of MPs emitting the field in Figure 2. Red isoregions are drawn to further highlight the emerging structure of the field.

4.3. Chemical Sensing

Each MP has eight receptors evenly distributed on its surface. The placement of the receptors, within the MP’s local coordinate system, begins at $45^\circ$ in the upper right and proceeds clockwise. The overall chemical gradient sensed by an MP is calculated with the following equations,

$$C_x = \frac{\Lambda_2 - \Lambda_6 + \frac{\Delta_3 - \Delta_8 - \Delta_1 + \Delta_4}{\sqrt{2}} \cdot r_c}{2 \cdot r_c}$$

$$C_y = \frac{\Lambda_8 - \Lambda_4 + \frac{\Delta_2 + \Delta_1 - \Delta_5 - \Delta_3}{\sqrt{2}} \cdot r_c}{2 \cdot r_c}$$

$$\nabla C = (C_x, C_y)$$

Figure 2. Typical chemical field emitted by an MP.
where $\Lambda_i$ is the chemical concentration calculated at receptor $r_i$, and $r_c$ is the cell’s radius. The locations of the numbered receptors are provided in Figure 4.

### 4.4. Parameters of Chemical Field Function

The chemical fields are defined as a function of the distance $d$ between the surfaces of two MPs, the angle $\theta$ between two MPs measured in an MP’s local coordinate system, as well as the MP’s age $t$. We define the chemical fields as mathematical functions with **Function set**: $F = \{+, -, \ast, \div, \exp, \log, \sin, \cos\}$ and **Terminal set**: $T = \{E(constant), d, t, \theta\}$. We also utilize a number of protected operators, such as protected division and protected logarithm. These protected operators safely handle invalid input values, for example divide by zero and non-positive logarithms.

The calculation of $d$ is straightforward, and involves calculating the distance between two 2D points (the receptor on one MP to the center of another) minus $r_c$, while taking into account the toroidal environment. In our system, $\theta$ ranges between $0^\circ$ to $90^\circ$ and is calculated within each MP’s local coordinate system. Therefore, we define symmetric values of $\theta$ as illustrated in Figure 5 with the HSV color scheme (red: $0^\circ$ to blue: $90^\circ$) in the local coordinate system of the grey cell in the image center. All MPs have the same age, which is the simulation time $t$ and is increased by 60 seconds of simulation time in each time step.

### 4.5. Movement

The “chemical” gradient is used to determine the primitive’s velocity. We assume that MPs travel at a terminal velocity through a viscous fluid environment, therefore an MP’s velocity is directly proportional to the chemical field gradient ($\nabla C$). When an MP moves in the direction of the...
chemical gradient, its velocity is calculated as

\[ \text{Velocity} = \lambda \ast \nabla C, \]

where \( \lambda \) (1 for our examples) is a constant that determines the magnitude of a cell’s response to the gradient. At each simulation time step \( (\Delta t) \) the displacement of the MP is

\[ \Delta x = \text{Velocity} \ast \Delta t. \]

If the displacement makes the MP collide with another MP, a small random step is taken instead. MPs do not always follow the field gradient. 10% of the time MPs take a random small step instead of following the field gradient. This randomness injects a small amount of noise into the system, helping to prevent the set of MPs from collecting into local minima configurations.

MPs also align with nearby neighbors. At first, MPs are randomly placed in the environment with random orientation \( \omega \), which ranges from \( 0^\circ \) to \( 359^\circ \). In our simulations, the “nearby” neighbors are those MPs within a four cell radii distance. Each MP identifies its nearby neighbors, detects its neighbors’ orientations, and calculates the average orientation of these neighbors \( \bar{\omega} \) by a vector average of the neighbors’ orientation. To perform alignment, MPs rotate one degree along the shortest arc towards the average orientation of its neighbors.

\[
\hat{\omega}_i = \cos(\omega_i) + i\sin(\omega_i),
\]

\[
\hat{\omega} = \frac{1}{n} \sum_{i} \hat{\omega}_i = x + iy,
\]

\[
\bar{\omega} = \begin{cases} 
    \arccos\left(\frac{x}{||\hat{\omega}||}\right) & y \leq 0, \\
    360^\circ - \arccos\left(\frac{x}{||\hat{\omega}||}\right) & \text{otherwise}.
\end{cases}
\]

\[
\omega_{\text{new}} = \begin{cases} 
    \omega + \frac{\hat{\omega} - \omega}{||\hat{\omega}||} & ||\hat{\omega} - \omega|| \leq 180^\circ, \\
    \omega - \frac{\hat{\omega} - \omega}{||\hat{\omega}||} & \text{otherwise}.
\end{cases}
\]

\( \hat{\omega} \) is vector presentation of MP’s orientation \( \omega \) and \( \omega_{\text{new}} \) is the new orientation after the MP has rotated towards the average of its neighbors’ orientations.

5. MP Self-Alignment and Self-Organization

Numerous experiments and simulations were run with our system. Our results demonstrate two aspects of emergent behavior, shape composition and intrinsic emergence.

5.1. Shape Composition

The shape composition feature of our system is an extension of pattern formation. Giving all MPs the same orientation \( (\omega = 0^\circ) \) and not allowing them to rotate, our system can be used to discover chemical field functions (i.e. local interaction rules) that direct primitives to form into user-defined macroscopic shapes [2]. In the shape composition process, since MPs do not rotate, they do not perform self-alignment. Figure 6 shows many of the results from this process.

5.2. Intrinsic Emergence

By allowing MPs to rotate, our system demonstrates that MPs not only form into certain shapes but can also perform alignment. All MPs are randomly placed in the environment with random initial orientations. The orientation may be indicated with the HSV color scheme, with \( \omega = 0^\circ \) representing red, \( \omega = 120^\circ \) representing green, and \( \omega = 240^\circ \) representing blue. Each MP is a colored disk, with a pink bar representing its orientation. The horizontal right direction (+x) is defined as \( \omega = 0^\circ \). MP self-alignment is demonstrated in Figure 7 through Figure 10. MPs tend to take on a color (i.e. orientation) similar to their neighbors’. The black disk at the center of the image indicates the center of mass of the aggregate.
Figure 7. MPs self-aligning and self-organizing into an ellipse.

Figure 8. MPs self-aligning and self-organizing into a gear-like shape.

Figure 9. MPs self-aligning and self-organizing into an hourglass shape.

Figure 10. MPs self-aligning and self-organizing into stripes with different alignment.
Figure 11. Genetic programming process that produces functions for morphogenetic primitives.

6. System Implementation and Analysis

6.1. Distributed Genetic Programming

The general genetic programming approach to defining field functions that ultimately produce the user-desired shape is presented in Figure 11. We start with a population of functions, which is initially randomly generated. Each function is compiled into a chemotaxis-based cell aggregation simulation program, and defines the chemical field that surrounds the individual cells. A simulation program is executed for each field function on a node in our cluster, usually producing some kind of aggregated structure. The resulting MP configuration is compared to the user-desired shape, and a scalar fitness value is calculated that quantifies how well the computed shape matches the desired shape. A subset of the top candidates are then used to create the next generation of field functions. The process continues until a field function produces the desired shape or the maximum number of generations is reached.

Since MPs are not aware of a global coordinate system, we do not want the shape of the resulting aggregate to be tied to a specific fixed orientation. We therefore calculate and align the centroids and major axes of the aggregated and target shapes. This allows MPs to aggregate into the desired shape, but not necessarily in the same position and orientation of the target. Once the global alignment is completed, the fitness function calculates the ratio of the overlapping pixels of the aggregate shape and the target shape to the total pixels of the target shape.

\[ f = \frac{\sum p_i, p_i \in S_a \cap S_t}{\sum p_j, p_j \in S_t}, \]  

where \( f \) is the fitness of the individual, \( p_i \) are the shape pixels that are present in both the aggregate and target image, \( p_j \) are the shape pixels in the target image, and \( S_a \) and \( S_t \) are the sets of non-white pixels in the aggregate and target shape images.

We have implemented an N-slave, 1-master, steady-state genetic-programming model, with the master process adaptively distributing individuals among the slave processes [1], based on Unix shell scripting [5]. Open Beagle [17], a C++ evolutionary computing framework, has been utilized and altered to function as the distributed genetic programming framework. Since the overhead of distributing the computation over the nodes is quite small compared to the simulation times themselves, we found that we achieved nearly an \( 8 \times \) speed-up by performing the calculations on an 8-node cluster. Given that each aggregation simulation requires approximately 4-CPU minutes, each generation consists of 100 individuals and that we usually produce the desired function by the 20th generation, we have found that we can derive a shape-specific field function in approximately one day.
6.2. Robustness and Scalability

In order to evaluate the reliability/robustness of our approach, 100 aggregation simulations each were performed for the ellipse, diamond, hourglass, boomerang and gear shapes. The repeatability for the ellipse was 100% (i.e. every aggregation simulation produced an ellipse-like shape), the diamond was 99%, the hourglass shape was 10%, the boomerang shape was 4% and the gear shape was 34%.

We are currently exploring the scalability of the system. We generated the ellipse shape with different numbers of MPs in the simulation. Figure 12 shows the MPs’ emergent behavior under the same local interaction rules, but with varying numbers of primitives. If the number of MPs falls below a certain threshold (for example, 400 in this case), they no longer aggregate into the ellipse shape. Note that because of the unrestricted orientation of the MPs, the major axis of the aggregate may have an arbitrary direction. Because all MPs have finite influence in the environment, there is a correlation between the overall density of the MPs and number of MPs that influence each other. When the number of MPs drops below a certain threshold, 400 in our example, there are an insufficient number of MPs to produce the emergent behavior. These initial experiments indicate the need for future investigations that will provide insights into the relationship between the system’s parameters and the desired outcomes.

7. Future Work

Motivated by the cell-level processes that create biological structures, we have presented chemical gradients (chemotaxis) as the core conceptual framework for the interaction of Morphogenetic Primitives (MPs). In the future, we would like to explore other mechanisms of morphogenesis [9] (shape, motility, friction (durotaxis), contact/adhesion (haptotaxis), cell division (proliferation) and cell death (apoptosis)) in order to produce more complex self-organizing shapes.

We also intend to further explore the intrinsic emergence aspects of the system. We have shown that MPs perform self-alignment and self-organization with local interactions, in that MPs aggregate into pre-defined shapes and exhibit certain global information through local alignment. We would like to further explore how this alignment can lead to the creation of more intricate structures. We are also interested in studying/developing the self-adaptation aspects of the system. For example, MPs can divide or die in the emergent process. Will the same local interaction rules suffice in directing MPs into forming pre-defined shapes under these conditions? In the long run, we would like to develop more elaborate MP behaviors, as well as extend the simulation model to 3D.

8. Conclusion

Motivated by the natural phenomenon of living cells self-organizing into specific shapes and structures, we have presented an emergent system that utilizes evolutionary computing methods for designing and simulating self-aligning and self-organizing shape primitives. Given the complexity of the emergent behavior, genetic programming is employed to control the evolution of our emergent system. The system has two levels of description. At the macroscopic level, a user-specified, pre-defined shape is given as input to the system. The system outputs local interaction rules that direct morphogenetic primitives (MP) to aggregate into the shape. At the microscopic level, MPs follow interaction rules based only on local information. All MPs are identical and do not know the final shape to be formed. The aggregate is then evaluated at the macroscopic level for its similarity to the user-defined shape. We have utilized the system to define MPs that form into a number of shapes, e.g. an ellipse, a diamond, an hourglass and a boomerang. Unexpectedly, a gear-like shape was also produced by the evolutionary process.
References


