A Manifold Operator Representation for Adaptive Design *

Jacob Beal
Raytheon BBN Technologies
Cambridge, MA, USA 02138
jakebeal@bbn.com

Hala Mostafa
Raytheon BBN Technologies
Cambridge, MA, USA 02138
hmostafa@bbn.com

Annan Mozeika
iRobot Corporation
Bedford, MA, USA 01730
amozeika@irobot.com

Benjamin Axelrod
iRobot Corporation
Bedford, MA, USA 01730
baxelrod@irobot.com

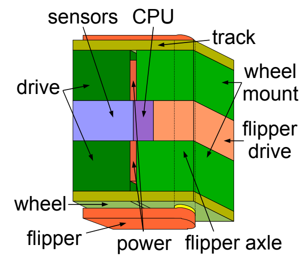
Aaron Adler
Raytheon BBN Technologies
Cambridge, MA, USA 02138
aadler@bbn.com

Gretchen Markiewicz
Raytheon BBN Technologies
Cambridge, MA, USA 02138
gmarkiew@bbn.com

ABSTRACT

Many natural organisms exhibit canalization: small genetic changes are accommodated by adaptation in other systems that interact with them. Engineered systems, however, are typically quite brittle, making design automation extremely difficult. We propose to address this problem with a generative representation of design based on manifold operators. The operator set we propose combines the intuitive simplicity of top-down rewrite rules with the flexibility and distortion tolerance of bottom-up GRN-based models. An embryogeny specified using this representation thus places constraints on a developing design, rather than specifying a fixed body plan, allowing canalization processes to modulate the design as it continues to develop. We demonstrate our ideas in the domain of electromechanical design and validate them with simulations at different levels of abstraction.

(a) miniDroid



(b) Partly developed embryo

Figure 1: We investigate embryogeny as a representation for adaptable structures using the example of a "miniDroid" robot (a), beginning by developing a body plan (b) from an undifferentiated egg.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques; I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Design

Keywords

Morphogenetic Engineering, Generative Design, Implicit Embryogeny, Generative Representation, Functional Blueprints

1. INTRODUCTION

When human engineers make modifications to a design, not all parameters are treated equally. Instead, a design expert typically begins by addressing only the parameters most critical to the desired modification. Constraints between the elements that are modified and other parts of the design then imply other changes, which may ripple throughout the entire design. In fact, human engineers consider many of these implicit changes to not be changes at all, but in fact to be *keeping the same design*.

This suggests that design evolvability by any automated system, from evolutionary algorithms to knowledge-based design tools, might be greatly enhanced by using a representation capturing relations exploited by a human designer. This would allow each element to adjust its phenotype for compatibility with other elements it interacts with.

In natural biological organisms, the processes of morphogenesis adapt structure to environment remarkably well on both an individual and evolutionary time scale. Significantly for our purposes, small genetic or somatic changes can produce cascading effects during development in other systems that interact with them. This ability to accommodate small changes without causing related components to fail is termed *canalization* [29]. Great strides have been made in understanding the building blocks of biological adaptivity and the ties between morphogenesis and evolution [9, 19], and there have been a number of systems that apply ideas from morphogenesis to particular aspects of design (see Section 2). No clear framework has yet been developed, however, for exploiting morphogenetic principles in the creation of general engineered systems.

In this paper, we present a generative representation for

*Sponsored by DARPA under contract W91CRB-11-C-0052; views and conclusions contained in this document are those of the authors and not DARPA or the U.S. Gov't.

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.

electromechanical system designs, using as an example a small ground robot, the "miniDroid" (Figure 1). Generative representations have the advantages of being compact and capable of representing highly complex phenotypes. Such representations are typically implemented either as top-down rewrite rules or bottom-up cellular models. The former has the advantage of being more intuitive to craft, while the latter exhibit more tolerance to distortions. Our encoding has the best of both worlds because it is rule based, but the rules invoke manifold operators that are robust to distortions (executing correctly regardless of the geometry).

Our approach promotes canalization; the result of our embryogeny is not the final body plan, but a web of constraints and relations among design parameters. This might be applied to both the form of a body and its controller, though in this paper we restrict ourselves to considering only the form. This affords more freedom to the design process to adapt the design by changing key parameters and having the rest be derived in accordance with the constraints. We suggest that this may allow for more flexible and efficient designs, where much of the process of adaptation can be carried out autonomously.

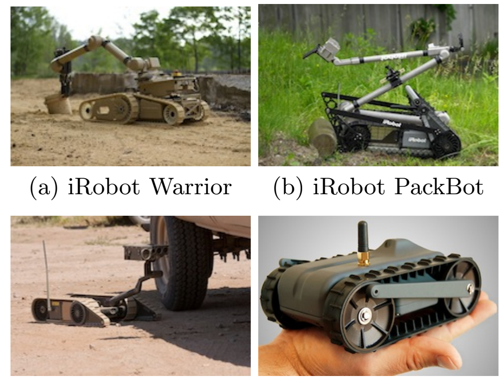
We instantiate our morphogenetic model framework with a candidate basis set of five manifold geometry operators, inspired by common processes in the embryogeny of animals. Such programs might then be integrated together with functional blueprints [1, 3] to facilitate design adaptation. Finally, we use the miniDroid for preliminary validation of this approach, creating a program for development of a robot body plan and testing the approach with simulations at several levels of abstraction.

2. RELATED WORK

Automated design of electrical and mechanical systems has long been a topic of interest. While there has been much work in the area (e.g., [8], [16], [14]), it has faced significant obstacles from the complexity of searching the space of possible designs. A primary aim of our work is a principled reduction of this design space.

This aim matches the goals of "morphogenetic engineering" as proposed by Doursat [12] and for robotics in [18] and [28]. Initial work towards a framework for morphogenetic engineering has been laid out for evolvable pattern formation in [11]. Meng et al.'s morphogenetic approach [26] to modular robotics combines a rule-based controller to generate desired patterns with a simulated genetic regulatory network to coordinate the configuration of robot modules. A more formal mathematical model can be found in [25], though the representational consequences are not explored.

An embryogeny is the process of growth that defines how a genotype is mapped to a phenotype [6]. Several works advocate the use of indirect mappings (also known in the Evolutionary Computing literature as implicit embryogeny) from genotype to phenotype (e.g., [6], [15], [17]). The genotype in this case is a generative encoding of the rules that govern the development of an individual. Generative encodings have the advantages of being compact and capable of representing highly complex phenotypes. This is due to parts of the genome being reused during development across time and space. Implicit embryogeny is implemented either as top-down rewrite rules or bottom-up cellular models. In subsequent sections, we elaborate on the similarities between our approach and each of these two approaches.



(a) iRobot Warrior (b) iRobot PackBot

(c) iRobot SUGV (d) iRobot LANdroid

Figure 2: Families of engineered systems often exhibit "phylogenetic" relationships similar to those of natural organisms.

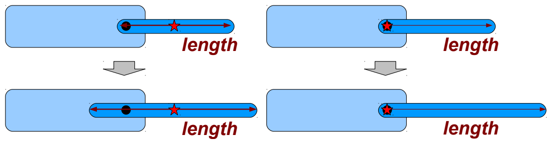
A number of specialized languages for shape formation have been inspired by morphogenesis. For example, chemical gradients are the basis of Nagpal's Origami Shape Language [27], which produces deformable geometric patterns, and Coore's Growing Point Language [10], which forms topological patterns using a model of plant tropisms. Likewise, Kondacs' model of pattern formation [21] creates regenerable patterns using a model of cell reproduction and apoptosis. However, such languages have been specialized and hard to apply to the design of realistic engineered systems.

Regardless of the specifics of the representation, genetic algorithms are typically used as the design process that modifies a genome to achieve a desired form/function. An alternative is offered by functional blueprints [1, 3], an engineering approach in which a system is specified according to its desired performance, with programs to incrementally adjust structure to improve performance when the goals are not met. The representation we provide in this paper is well-suited for interaction with either genetic algorithms or functional blueprints.

3. EMBRYOGENY CAN ENCODE DESIGN PARAMETER RELATIONS

In any engineered system, some parameters strongly influence overall system design, while others can be derived by an expert human designer from the interaction of the key parameters. To better understand how these sort of systematic relations might be exploited, we focus on the example of electromechanical systems, and in particular the the family of products that the iRobot corporation has derived from their original PackBot system (Figure 2), and a new "miniDroid" design based on the same architecture (Figure 1(a)). Although these robots span a wide range of sizes and applications, all of them share a base body plan, including paired treads each driven by two wheels, flippers coaxial with one wheel, and a top-mounted sensor/manipulator package. The loose similarity to natural phylogenetic families suggests that the natural world may yield hints toward a design representation that facilitates such variation.

Consider a simple modification to the miniDroid which, like all robots in the PackBot family, uses its flippers to climb over obstacles. When a designer increases the miniDroid's



(a) Part-Centric (b) Interface-Centric

Figure 3: Coordinate systems can implicitly encode design: interface-centric coordinates let a miniDroid flipper extend outward when lengthened.

flipper length (e.g., to climb over slightly taller obstacles) should the flipper extend symmetrically around its geometric center (Figure 3(a)), away from its attachment point (Figure 3(b)), or some combination thereof? The answer is intuitive to a human designer, but a self-adapting design must have some way of representing this relationship.

This is potentially a severe problem: engineered designs have extremely high numbers of parameters: even a simple beam is described by at least nine parameters (side lengths, position, and orientation). Only a few of these parameters are key, while most are constrained by their relationship to key parameters. Natural systems, however, must solve this problem in the embryogenies that create their forms, since evolution acts through relatively independent incremental changes to very few parameters at a time.

In animals, the process of embryogenesis effectively specifies a hierarchy of spatial relationships and coordinate systems that constrain how changes can occur. For example, when an arm is lengthened, it extends further out from the body, rather than attempting to invade into the body.

We thus look to embryogeny as a means of encoding relationships between design parameters, defining it as a partially ordered sequence of operations over a spatial computer that convert a simple initial "egg" into a "mature" structure. It is important to note that although the systems we design are not going to be physically built using a developmental process, we still need to simulate this process because it generates by-products (e.g., various coordinate systems local to tissues, tissue specialization hierarchies, and simulated physical interactions between tissues) that guide the maintenance of a design's integration as its components are being changed, whether by a human designer, an evolutionary algorithm, or an online process for repair and adaptation.

4. DEVELOPMENTAL REPRESENTATIONS

We now propose a representation of embryogeny using a candidate basis set of manifold geometry operators, then demonstrate how this can be applied to develop the body plan of a miniDroid. For reasons of space, we give only a high-level description of the operators and the implementations we have constructed. Full details can be obtained from the MADV release online at `http://madv.bbn.com`.

4.1 Manifold Operators

In natural systems, a vast array of mechanisms are used to develop the form of an organism. From this panoply, we extract five simple abstract manifold geometry operators for measuring, segmenting, and modifying manifolds. This is not necessarily the "best" set of operators, but for preliminary work, we minimize the complexity of the model in order to simplify analysis of its capabilities and implications.

We formulate these as operators on continuous manifolds, rather than simpler Euclidean geometric operators, so that they can tolerate topological distortions. Thus when a structure's shape and size are changed (either by changes to its antecedents or by physical interaction with other structures), the operators should still produce qualitatively similar effects, promoting canalization. These operators can be applied to either the entire structure or to any substructure (viewed as a submanifold), and affect surrounding tissues only indirectly, as a byproduct of physical interaction.

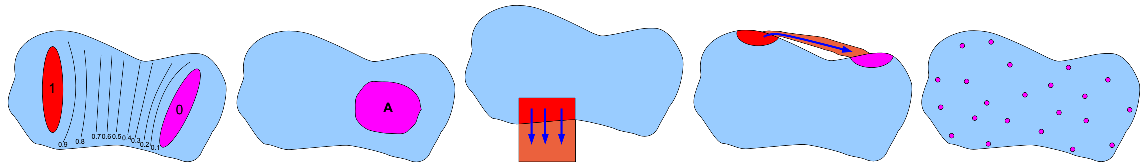
Figure 4 illustrates our set of manifold operators:

- **coordinatize(0-region, 1-region)**: This operator computes a local coordinate system between two regions, ranging from 0 in one region linearly to 1 in the other region. It is based on the use of chemical gradients to polarize tissues, and most particularly on those where the signal is the ratio of two chemicals.
- **latch(region, type)**: The latch operator differentiates a region into a component type. This operator is analogous to determination of cell fate in a biological system, such as the determination of limb fields.
- **scale(region, scale-factor)**: The scale operator takes a region and increases or decreases its scale proportionately along one or more axes. This is inspired by directed proliferation, such as is used in the construction of limb buds.
- **connect(source-region, destination-region)**: This operator extends a new region from a source region along a near-shortest path to a destination region. It is based on cell migration, particularly the migration of tissue sheets, and on the extension of cell processes such as axons. We have tentatively specified that only the closest point in the source will connect, and that the width of the connection will be proportional to the size of the smaller region.
- **speckle(region, expected-separation)**: The speckle operator selects a set of small regions filling the space, each region determined randomly and separated by an expected distance from all others nearby. This is inspired by symmetry-breaking processes, such as those used to create hair follicles.

When composed using arithmetic, branching, and function definition, we believe that this set of manifold operators may be sufficient to express most arbitrarily complex adaptable structures (though they are not space-time universal [2]). Many other biological mechanisms can also be expressed as composites of these operators: for example, recruitment can be expressed as coordinatize followed by latch.

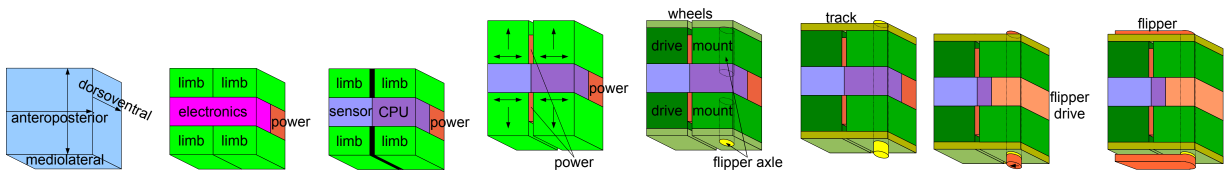
4.2 Example: miniDroid Body Plan

To validate the plausibility of our candidate operators, we have created a draft developmental sequence for the body plan of a miniDroid, which is likely to also apply well to other members of the PackBot family. The sequence begins with a cuboid egg (to better match current engineered systems and manufacturing processes), and uses all operators except for speckle. Our developmental sequence can be viewed in terms of eight stages, shown in Figure 5, though execution may proceed somewhat asynchronously:



(a) Coordinatize (b) Latch (c) Scale (d) Connect (e) Speckle

Figure 4: Candidate manifold geometry operators to be a basis set for developing electromechanical designs.



(a) Stage 1 (b) Stage 2 (c) Stage 3 (d) Stage 4 (e) Stage 5 (f) Stage 6 (g) Stage 7 (h) Stage 8

Figure 5: Approximate developmental stages of a miniDroid (or other PackBot family member) body plan.

- Stage 1: basal coordinates are created, along anteroposterior, dorsoventral, and mediolateral axes. **Operators: coordinatize**
- Stage 2: basal coordinates are used to partition the robot into coarse body plan, with the exterior latching into "skin." **Operators: latch**
- Stage 3: the electronics region differentiates along the anteroposterior axis into sensor and CPU regions. The boundary between limb buds recruits nearby material to form a gap between anterior and posterior limbs. **Operators: coordinatize, latch**
- Stage 4: limb buds scale themselves out of the body, then establish a proximodistal axis and local centromedial axis. The power region recruits inter-limb gap material. **Operators: coordinatize, latch, scale**
- Stage 5: distal section of limb buds differentiates into wheels. Proximal section of front limb buds differentiates into drive. For rear limb buds, the flipper axle differentiates using the centromedial axis, and the proximal section of the limb bud differentiates into a passive mounting point. The wheels will round themselves later, as the body scales up. **Operators: latch**
- Stage 6: flipper axles scale in and out, meeting in the center. Wheel edges connect tracks between wheels. **Operators: scale, connect, latch**
- Stage 7: flipper axles scale outwards to form the flipper drive, pressing away nearby sections of CPU and power. Distal sections of axle differentiate into flipper buds and form a new anteroposterior axis. **Operators: scale, coordinatize, latch**
- Stage 8: flippers scale from buds. **Operators: scale**

At Stage 8, the complete body plan has been formed. Each of the sections can now refine its details, while scaling up to reach the mature size and component scale relations specified by its genotype. The connectivity relations developed in the body plan constrain this scaling process, resolving implicit parameters: for example, if the flipper grows longer it can only do so by lengthening the portion not attached to the axle, and if the overall body grows wider, the wheel and motor assemblies will be carried outward along with the expansion. Meanwhile, power and signal wires connect the CPU, batteries, and motors using the connect operator, a process similar to innervation in vertebrates.

Together, these processes should produce a design similar to that of the miniDroid. Integration with a (semi)autonomous design process, such as an evolutionary algorithms or functional blueprints, should be possible by placing the scaling parameters and/or embryogeny under control of the design process.

5. IMPLEMENTATION

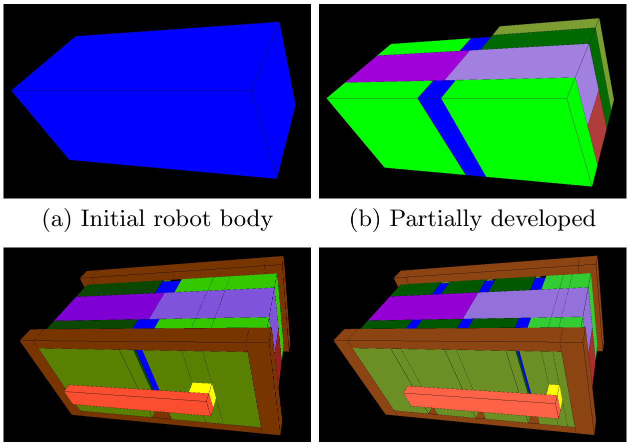
To begin validation of our approach, we have implemented simulations at four different levels of abstraction:

- At the highest level of abstraction, a tissue-level simulator demonstrates flexible application of manifold operators using an asynchronous rule system.
- Refining to a cellular model, we demonstrate that using manifold operators allows a design to maintain coherent inter-component relations even if the adaptation process subjects it to severe distortions.
- A soft-body simulation demonstrates the ability to compute adhesion, displacement, and penetration of components at reasonable computational cost, enabling physics-based resolution of component layout.
- Finally, adaptive refinement of details under feedback control is demonstrated with a chemotactic wiring model.

Each of these demonstrates a different aspect of how our morphogenetic approach enables canalization of a developing design. Having demonstrated that each requirement of the system is individually reasonable, the next obvious step will be to integrate them to produce a single system that completely constrains the realization of design phenotypes.

5.1 Tissue-Level Models

In the previous section, we proposed a set of manifold operators for representing development. The tissue-level model



(a) Initial robot body (b) Partially developed

(c) Four wheeled robot (d) Eight wheeled variant

Figure 6: (a) The initial robot body is (b) updated using the set of development rules to create either (c) a four or (d) eight wheeled robot (wheels are unrounded green boxes).

demonstrates how these can be combined into a flexible embryogeny through asynchronous rule execution.

For this simulation, we implemented a prototypical specification of embryogeny, executor, and visualizer in Java. The simulation uses a coarse-grained representation of cell collectives (tissues), where all cells carry out the same function, analogous to biological tissues. Each tissue contains a set of body part labels (e.g., "flipperBud"), a geometry specifying its shape, and a means of converting between its own coordinate system and others.

The tissues are manipulated by a set of developmental rules. Our choice of asynchronous rules for a representation is inspired by naturally occurring genetic regulatory networks (GRNs): both of these have the useful property of separating execution conditions (promoter sequence of GRNs, preconditions of rules) from actions (encoded protein/nucleic-acid complexes of GRNs, post-conditions of rules). The rules are thus loosely coupled, reducing the need for explicit ordering constraints on their execution. As in other generative design systems, iteration and conditional execution are implicit and arise out of the interaction of the rules.

In our representation, a rule applies to any tissue that matches the rule's preconditions. An executor starts firing the developmental rules on an undifferentiated robot body ('egg'), breaking ties in arbitrary order. As rules differentiate and manipulate parts of the body, other rules become applicable. A rule's effects are a sequence of manifold operators from the basis set given in 4.1.

A difference between our developmental rules and traditional re-write rules is that ours operate in the context of a physics simulator that places physical constraints on their effects. Another important difference is that our rules can have non-local triggers and effects (e.g., a rule involving scaling can result in attached tissues being dragged along).

We have implemented a miniDroid embryogeny, which uses 12 rule skeletons that are parameterized into 23 rules that carry out the 8 developmental stages described in the previous section. Currently, our rules avoid operations that would require interactions between tissues, e.g., penetrating other tissue. Our work to support such interactions is described in Sections 5.2 and 5.3. The rules produce a body plan of a robotic "embryo" made up of various "organs" like limbs, flippers and wheels. We were careful to keep the rules oblivious to absolute dimensions and positions, a task that was facilitated by the use of the manifold operators. Figure 6 shows the initial body (6(a)) and the body-plan after embryogeny (6(c)). An example of the adaptability of our encoding is shown in Figure 6(d), which shows an eight-wheeled variant created by changing the single parameter specifying the number of body segments (pairs of wheels).

The body plan is not yet the final design, but effectively encodes constraints on the design parameters, allowing many implicit parameters to be resolved by means of the geometric and topological relationships established in the elaborated manifold developed by executing these rules. For example, the CPU and batteries are inside the body, though their relative positions within are not fixed, and the flippers are attached at the axle, so if they are made longer, they can only grow outward away from the axle.

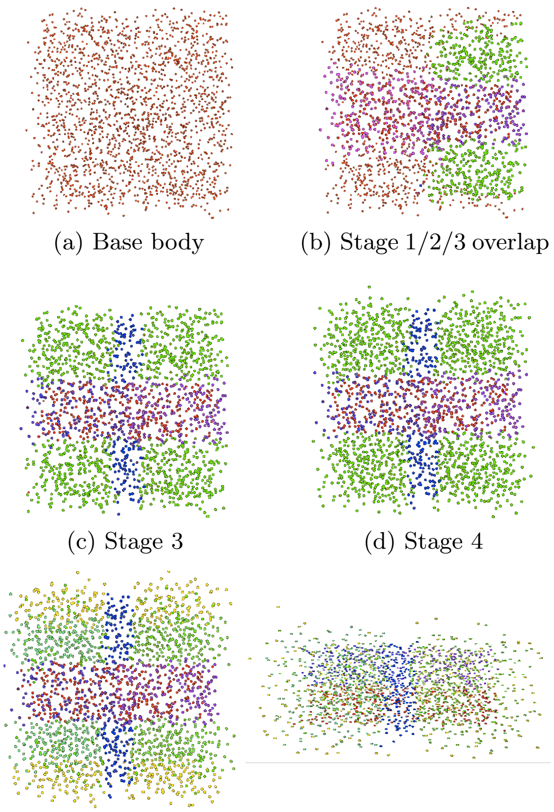
Although rule-based generative encodings have been widely used (e.g., [17], [20], [23]), we note that our encoding and the resulting phenotypes are very different from those of prior systems. Existing approaches develop systems consisting of large numbers of simple repeated structures. For example, in Framsticks [20], the body of a robot is composed of a set of interconnected simple elements called sticks. The simplicity, uniformity and large numbers of these system building blocks allow the rewrite rules significant freedom in manipulating the structure. Most electromechanical designs, however, like the miniDroid, are made of parts that are highly heterogeneous and tightly coupled. Our rules therefore cannot arbitrarily rewrite the body, but need a representation like ours, which allows parts to maintain particular physical and topological relationships as they differentiate.

5.2 Cellular Models

Computing arbitrary distortion of manifolds at the tissue level is extremely challenging due to the difficulty of analytically representing and manipulating arbitrary geometries, so we demonstrate the distortion tolerance of manifold operators using a cellular simulation instead. We implemented this model in the Proto [4] spatial computing language. In Proto, a program is specified in terms of geometric computations and information flow on a continuous manifold, then compiled to generate a local program that approximates the global specification. Because our proposed manifold operators were deliberately selected to be compatible with such transformations, the local programs produced from them are simple and could be readily transformed into a GRN representation.[1] Proto thus links tissue-level representations, such as those discussed in the previous section, with cellular models where it is simpler to represent approximate topological distortions.

We have implemented the first five stages of the miniDroid embryogeny, from the initial coordinatization of the initial body through the scaling and differentiation of limb buds. While the Proto implementation does not represent rules (we instead transform the dependencies into flow control), its manifold representation allows the program to execute under distortion on a wide range of alternate manifolds—a

[1]In fact, Proto programs have already been transformed into actual biological GRNs [5].



(a) Base body (b) Stage 1/2/3 overlap

(c) Stage 3 (d) Stage 4

(e) Stage 5 (f) Stage 5, side view

Figure 7: Stages 1 to 5 of body-plan development executed in fine-grained Proto models. From the initial body (a), the spreading coordinate systems trigger differentiation (b) into the base components (c). The limb buds then scale outward (d) and differentiate into wheels, motors, and mounts (e,f).

critical capability for emulating the adaptivity exhibited by natural morphogenesis. The continuous model of distributed execution also provides a natural path toward using blending to resolve conflicts between rules.

Figure 7 shows snapshots of embryogeny executing on a model of 2000 cells distributed through a rectilinear 3D volume of space. Figure 8 compares the result of execution the same program with different initial conditions, showing the inherent geometric adaptation of the program.

Notice that the operators of the embryogeny are able to execute both asynchronously and simultaneously, and that the manifold representation allows the program to produce sensible results even on a radically distorted underlying manifold. The mathematical nature of manifold geometry thus provides one form of canalization: whenever a region of space is transformed, the coordinate fields created by coordinatize operators transform along with it, and any subsequent operator that is applied to that region is transformed to match as a consequence.

Although our model is "cellular," it is very different from the cellular models discussed in works on implicit embryogeny, which tend to be more bio-mimetic, capturing low-level biological processes like GRNs and cell-level interactions (e.g., the work by Eggenberger [13] implements concepts like cell division, death and induction at the cell and GRN level). The question often addressed in the literature is how to go from low-level interactions to the desired global behavior/structure. The answer typically relies on an external element (e.g., fitness function in a genetic algorithm) to assess the collective behavior/structure arising out of the interactions. This assessment is then used as a guide to make changes to the low-level interactions that hopefully (though not directly) move the system closer to its goal. Our approach differs in that in our cellular model, we compile the desired global behaviors into low-level interactions, instead of the other way around. This greatly simplifies the problem, allowing us to develop more complex morphologies than those in the literature (e.g., [7], [13], [22]).

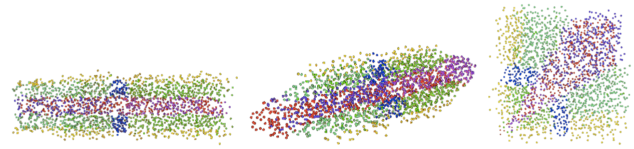


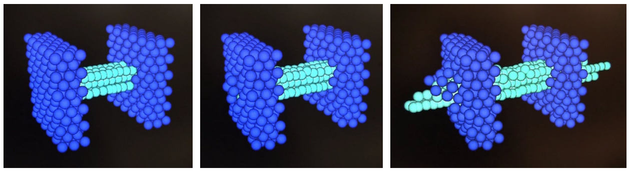
Figure 8: Proto's manifold model allows embryogeny to automatically adapt to execute on changed underlying spaces. Shown (l to r) are execution on a flat egg, an ovoid egg, and twisted coordinates.

5.3 Soft-Body Tissue Interaction

In many cases, developing components may try to move relative to connected components or to move into occupied space. The adaptability of our manifold representation allows many such conflicts in design layout to simply be resolved physically, by having tissues push, pull, shear, and penetrate one another, much like biological tissues. This promotes canalization by allowing structures to develop in a dynamically adjusted physical context (using the distortion tolerance discussed in the previous section), rather than forcing all geometric conflicts to be resolved in the developmental program. For example, in Stage 6 of miniDroid development, the flipper axles scale outward, penetrating the wheels (Figure 5(f)), which means that the wheel program need not be modified to allow for the development of flippers.

Implementing this interaction requires a soft-body simulation, which must be computationally inexpensive enough to be practical. We are not attempting to simulate tissue interactions in a way that is necessarily biologically plausible. We need only to represent tissues in a way that allows them to flexibly interact during growth operations.

As a first case study we simulated a tissue scaling operator penetrating into an adjacent tissue with the focus being on how the penetrated tissue responds (Figure 9), analogous to the flipper axle/wheel interaction described above. The axle must push up against the inner walls of the wheels; the wheel gives way elastically up to a point and then begins to plastically deform, allowing the flipper axle to penetrate. After investigating several representations of tissues, node based tissue representations have produced the most desirable results. Nodes (cells) were uniformly distributed throughout the volume contained by a surface mesh (initial state). The cells are outfitted with several interactions which allow the tissue to retain shape when under no external forces (from



(a) Initial State (b) Deformation (c) Penetration

Figure 9: Soft-body model of a shaft growing axially between two wheels (a), that deform outward (b), then are penetrated and deform plastically (c).

adjacent tissues) while allowing deformation to occur (plastically and elastically) when external forces are applied. A low resolution representation of the miniDroid has been implemented using the node based tissue representation.

Simulations were implemented using MASON, a Java-based multi-agent simulator [24]. Our interaction models use directed growth, implemented by assigning a growth vector $\boldsymbol{\lambda}$ to each class of cells (tissue type). The growth rate for each cell is computed from a neighborhood vector $\boldsymbol{\eta}$, which we define as the distance-weighted centroid of the N cells within distance M of the position of the current cell:[2]

$$\boldsymbol{\eta} = \sum_{i}^{N} \frac{\mathbf{x} - \mathbf{x_i}}{||\mathbf{x} - \mathbf{x_i}||} \tag{1}$$

where $\mathbf{x}$ is the position of the current cell, $\mathbf{x_i}$ is its i^{th} neighbor, and $||\ ||$ denotes vector length. The growth, or aging, rate for a particular cell, Γ, is the dot product of $\boldsymbol{\eta}$ and $\hat{\boldsymbol{\lambda}}$:

$$\Gamma = \boldsymbol{\eta} \cdot \hat{\boldsymbol{\lambda}} \tag{2}$$

where $\hat{\boldsymbol{\lambda}}$ is the normalized vector. A cell splits to create a child when its age reaches a certain threshold. The child is placed at a random location in a sphere of radius $\boldsymbol{\lambda}$, centered around the parent. Then the parent's location adjusted so that their centroid of the pair matches the parent's original location.

Directed growth is not enough to maintain tissue shape, so intercellular interactions are also directed according to $\boldsymbol{\lambda}$. We modeled three inter-cellular forces: 1) a strong repellent force when two cells are overlapping, 2) a weak attractive force between cells of the same class within some small distance, and 3) a weak repellent force between cells of different classes within some small distance. All three of these forces are governed with the same underlying equation:

$$\mathbf{f_{type_{ij}}} = (\mathbf{x_i} - \mathbf{x_j}) \frac{-f_{max_{type}}}{(r_i + r_j)} + f_{max_{type}} \tag{3}$$

where r_i and r_j are the two cell radii, and $f_{max_{type}}$ is the maximum value for that force type.

We found these forces to be necessary to keep groups of the same cell type together, and prevent too much mixing between types. These forces are then summed over neighbors as follows:

$$\mathbf{F} = \sum_{i}^{N} ((\mathbf{f_{1_i}} + \mathbf{f_{2_i}} + \mathbf{f_{3_i}}) \cdot \hat{\boldsymbol{\lambda}}) \frac{(\mathbf{x} - \mathbf{x_i})}{||\mathbf{x} - \mathbf{x_i}||} \tag{4}$$

[2]We found stability was improved by limiting the maximum number of neighbors in the calculation.

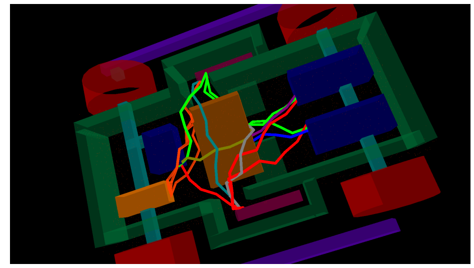


Figure 10: Wiring model executing in a simulated miniDroid interior. Components are modeled as rectangles, with wires routing around obstacles.

This ensures that motion due to inter-cellular forces is more pronounced in the directions that align with $\boldsymbol{\lambda}$.

Finally, we also ensure that cells that do not become too dense by killing off cells with high forces exerted on them. This also aids in deformation and penetration when cell classes collide.

Our simulation method is efficient, requiring only $O(nN)$ computation, where n is the number of cells and N the neighborhood size, but can still be much improved.

5.4 Wiring Models

The connect operator enables another form of canalization that is also quite important in biological organisms: the integration of systems that initially develop separately. An important example of this is the chemotactic process by which neurons grow axons to reach appropriate target cells in an organism's developing nervous system. Because this integration happens as a feedback process, perturbations such as congenital extra fingers or toes are simply integrated into the system and become functional.

With the miniDroid, wiring components to the CPU and power supply presents an analogous problem. We implemented a self-stabilizing connect operator in Proto, and used it to implement autonomous wiring design for the miniDroid. The connect operator is implemented by having its destination emit a chemical whose diffusion forms a gradient that propagates through penetrable tissue only (i.e., around solid obstacles like motors). The source follows this gradient back to the destination, forming a near-shortest path between them. Because the computation is self-stabilizing, when the structure changes the path automatically adjusts as well.

The wiring model for the miniDroid uses 9 connect operators to create 15 wires connecting the motors, CPU, batteries, and flipper encoder. Figure 10 shows the wiring model executing on a 4,000 cell approximation of a miniDroid interior, in which wires successfully connect signal, power, and ground between the batteries, CPU, encoder, and motors.

This demonstrates a fourth form of canalization enabled by our manifold operators, in which a functional blueprint acts during embryogeny to ensure functionality of a design by actively modifying it toward specified goals as it develops.

6. CONTRIBUTIONS AND FUTURE WORK

We have presented a morphogenetic engineering framework for simplifying design adaptation. This simplification is due to implicit design parameters being derived from key parameters, made possible by encoding the design constraints in our implicit embryogeny. We also proposed a basis set of manifold operators for representing distortion-tolerant embryogeny, and applied them to the example of the miniDroid. We further demonstrated the plausibility of this approach with four simulations at various levels of abstraction: models of tissue-level development, cellular development under distortion, soft-body component interaction, and wiring design. One measure of success of our design tool is comparing the size of the design space (e.g., in terms of the number of parameters a design process can manipulate) to the size of the space explored by our framework. Preliminary results show a dramatic decrease in the search space.

Three clear directions for future work follow from the work presented in this paper: first, integration of our simulations to produce a complete mapping from genotype to adapted phenotype. Second, integration with (semi)autonomous design adaptation methods such as evolutionary algorithms or functional blueprints, and quantification of the benefits of incorporating morphogenetic constraints. Finally, and on a more general level, theoretical investigation of how use of morphogenetic models modifies a fitness landscape.

7. ADDITIONAL AUTHORS

Additional authors: Kyle Usbeck (Raytheon BBN Technologies, email: `kusbeck@bbn.com`)

8. REFERENCES

[1] A. Adler, F. Yaman, J. Cleveland, and J. Beal. Morphogenetically assisted design variation. In *Int'l Conference on Morphological Computation*, 2011.

[2] J. Beal. A basis set of operators for space-time computations. In *Spatial Computing Workshop*, 2010.

[3] J. Beal. Functional blueprints: An approach to modularity in grown systems. *Swarm Intelligence*, 5(3), 2011.

[4] J. Beal and J. Bachrach. Infrastructure for engineered emergence in sensor/actuator networks. *IEEE Intelligent Systems*, pages 10–19, March/April 2006.

[5] J. Beal, T. Lu, and R. Weiss. Automatic compilation from high-level languages to genetic regulatory networks. *PLoS ONE*, 2011.

[6] P. Bentley and S. Kumar. Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In *GECCO*, 1999.

[7] J. C. Bongard and R. Pfeifer. Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In *GECCO*, pages 829–836, 2001.

[8] M. I. Campbell, J. Cagan, and K. Kotovsky. A-design: An agent-based approach to conceptual design in a dynamic environment. *Research in Engineering Design*, 11:172–192, 1999.

[9] S. B. Carroll. *Endless Forms Most Beautiful*. W. W. Norton & Company, 2005.

[10] D. Coore. *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer*. PhD thesis, MIT, 1999.

[11] R. Doursat. Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering. In R. Wurtz, editor, *Organic Computing*, pages 167–200. Springer-Verlag, 2008.

[12] R. Doursat. Morphogenetic engineering weds bio self-organization to human-designed systems. *PerAda Magazine*, 2011.

[13] P. Eggenberger. Evolving morphologies of simulated 3d organisms based on differential gene expression. pages 205–213. MIT Press, 1997.

[14] M. P. J. Fromherz, D. G. Bobrow, and J. de Kleer. Model-based computing for design and control of reconfigurable systems. *AI Magazine*, 24(4), 2003.

[15] T. Gordon and P. Bentley. Towards development in evolvable hardware. In *NASA/DoD Conference on Evolvable Hardware, 2002*, pages 241 – 250, 2002.

[16] S. P. Hoover and J. R. Rinderle. A synthesis strategy for mechanical devices. *Research in Engineering Design*, 1:87–103, 1989.

[17] G. S. Hornby and J. B. Pollack. Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*, 8:223–246, 2002.

[18] Y. Jin and Y. Meng. Morphogenetic robotics: An emerging new field in developmental robotics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(2):145–160, 2011.

[19] M. W. Kirschner and J. C. Norton. *The Plausibility of Life: Resolving Darwin's Dilemma*. Yale University Press, 2005.

[20] M. Komosinski. *Framsticks: a platform for modeling, simulating and evolving 3D creatures*, chapter 2, page 37–66. Springer, New York, first edition, 2005.

[21] A. Kondacs. Biologically-inspired self-assembly of 2d shapes, using global-to-local compilation. In *IJCAI*, 2003.

[22] S. Kumar and P. J. Bentley. Computational embryology: past, present and future. In *Advances in evolutionary computing: theory and applications*, pages 461–477. Springer Verlag (UK, 2003.

[23] A. Lindenmayer. Mathematical models for cellular interaction in development: Parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968.

[24] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. Mason: A multi-agent simulation environment. *Simulation*, 82(7):517–527, 2005.

[25] B. MacLennan. Models and mechanisms for artificial morphogenesis. In *Natural Computing, Proceedings in Information and Communications Technology*, volume 2, 2010.

[26] Y. Meng, Y. Zhang, and Y. Jin. A morphogenetic approach to self-reconfigurable modular robots using a hybrid hierarchical gene regulatory network. In *International Conference on the Synthesis and Siumulation of Living Systems*, 2010.

[27] R. Nagpal. *Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics*. PhD thesis, MIT, 2001.

[28] R. O'Grady, A. L. Christensen, C. Pinciroli, and M. Dorigo. Robots autonomously self-assemble into dedicated morphologies to solve different tasks. In *AAMAS*, 2010.

[29] C. H. Waddington. Canalization of Development and the Inheritance of Acquired Characters. *Nature*, 150(3811):563–565, November 1942.