

On the Relationship Between Environmental and Morphological Complexity in Evolved Robots

Joshua E. Auerbach
Morphology, Evolution and Cognition Lab
Department of Computer Science
University of Vermont
Burlington, VT 05401
joshua.auerbach@uvm.edu

Josh C. Bongard
Morphology, Evolution and Cognition Lab
Department of Computer Science
University of Vermont
Burlington, VT 05401
jbongard@uvm.edu

ABSTRACT

The principles of embodied cognition dictate that intelligent behavior must arise out of the coupled dynamics of an agent's brain, body, and environment. While the relationship between controllers and morphologies (brains and bodies) has been investigated, little is known about the interplay between morphological complexity and the complexity of a given task environment. It is hypothesized that the morphological complexity of a robot should increase commensurately with the complexity of its task environment. Here this hypothesis is tested by evolving robot morphologies in a simple environment and in more complex environments. More complex robots tend to evolve in the more complex environments lending support to this hypothesis. This suggests that gradually increasing the complexity of task environments may provide a principled approach to evolving more complex robots.

Categories and Subject Descriptors

I.2.9 [Computing Methodologies]: Artificial Intelligence—*Robotics*

General Terms

Experimentation

Keywords

Evolutionary Robotics, Morphological Complexity

1. INTRODUCTION

According to the principles of embodied cognition intelligent behavior arises out of the coupled dynamics between an agent's body, brain and environment [2, 6, 9, 26]. This suggests that the complexity of an agent's control policy

(brain) and physical body (morphology) should scale proportionally to the complexity of its task environment. This link between control and morphology has been studied [25], however the relationship between environmental complexity and morphological complexity is not well understood.

Evolutionary robotics (ER) [14, 23], the application of evolutionary algorithms to the design and optimization of robot control policies and/or morphologies, provides a framework for investigating this relationship. While most evolutionary robotics projects have restricted themselves to optimizing control strategies for human designed or bio-mimicked robot body plans, evolutionary algorithms may also be used to design complete robots: physical morphologies in addition to control policies. Evolving morphology in addition to control has the advantage of being able to discover body plans uniquely suited to a machine's given task environment rather than suffering from the design biases of human engineers.

This idea of allowing an evolutionary algorithm to control both the morphologies and controllers of simulated machines in virtual environments to produce adaptive behavior was first introduced by Sims [30]. Sims' was followed by other studies (e.g. [1, 3, 5, 7, 8, 13, 15, 17, 19, 21, 22, 32]) which also explored evolving both the morphologies and control policies of robots in virtual environments. These studies varied in a number of meaningful ways including their underlying genetic encodings, the parts with which the robots were constructed, the evolutionary algorithms employed, and the tasks investigated. However, by far the most commonly investigated task in this line of research has been locomotion over flat terrain: how far a robot is able to displace itself over flat ground in an allotted amount of time.

While interesting results have come from investigating this task it suffers from its simplicity. Relatively simple morphologies of just a few cuboids or spheres are all that is needed to be successful. However, it is of great interest how morphological complexity scales in more complex task environments, therefore additional task environments must be investigated. Previous studies have looked at evolving robots in more challenging task environments (e.g. [18]), but because these studies used body plans composed of cuboids, like Sims' system, there was a low ceiling on the maximum complexity of their evolved morphologies.

The current study aims to investigate the relationship between environmental and morphological complexity in a more principled way in order to test the hypothesis that the morphological complexity of a robot increases commensurately with the complexity of its task environment. While

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '12, July 7–11, 2012, Philadelphia, Pennsylvania, USA.
Copyright 2012 ACM 978-1-4503-1177-9/12/07 ...\$10.00.

it draws inspiration from the previous studies mentioned above, the evolutionary system presented here has several advantages which make it better suited to studying this issue.

One advantage concerns the genetic encoding employed and the manner in which robot morphologies are modeled. As has been demonstrated in the past [16, 17] generative and developmental encodings offer demonstrable benefits over direct encodings for evolving robot morphologies. Accordingly, the morphologies in this study are created from a specific generative encoding that has been shown to possess a host of advantages over other encodings: Compositional Pattern Producing Networks (CPPNs) [33]¹. This is similar to what was done in [3, 5], however in lieu of building robots out of spherical components via a growth procedure as is done in [3, 5] morphologies are instead created out of triangular meshes (trimeshes) based on sampling a CPPN output at regular intervals over a region of space. The flexibility of trimeshes allows for the creation of a greater diversity of morphologies than is possible with cuboids or spheres (see Figures 1 and 4 for examples of morphologies evolved with the current system).

Another advantage of the current system is the genetic algorithm employed. Many advances have been made in developing more successful evolutionary algorithms since Sims’ work, which should allow for searching the space of robot morphologies more effectively. Specifically, in this research, populations of CPPN genomes are evolved using CPPN-NEAT: an extension of the widely used NeuroEvolution of Augmenting Topologies (NEAT) [34] algorithm. Some of the advantages of CPPN-NEAT are presented in the next section.

A final advantage worth mentioning is the vast amount of computational resources that many modern researchers have access to. These resources are necessary to run large numbers of physical robotics simulations at small enough step sizes to produce physically plausible results. All the experiments presented in this paper are carried out on a 7.1 teraflop supercomputing cluster. Without access to such a distributed computing system one single evolutionary run from one single experiment would take multiple days to complete on a standard personal computer. But, when using the cluster, an entire experiment (of 100 runs) can be run in less than one day thus allowing for experimentation with a large number of environments within which enough runs may be conducted to produce statistically significant results.

The remainder of this paper is organized as follows: the next section further describes the CPPN encodings used, describes how they evolve and how they produce actuated robots. A description of the different simulated environments in which robots are evolved then follows. Next, results are presented which capture how different environments affect the complexity of the robot morphologies that evolve inside them. This is followed by a discussion of how the complexity of a robot body plan may be calculated using geometric properties and information theoretic measures. These techniques are then applied to the evolved robot body plans and relationships between environmental and morphological complexity are examined. The paper finishes with concluding remarks and a discussion of how the ideas presented in this paper may be extended in future work.

¹For more on the specific advantages of CPPNs as a genetic encoding the reader is directed to [3–5, 10, 12, 31, 33].

2. METHODS

2.1 CPPNs

Compositional Pattern Producing Networks (CPPNs) [33] are a form of artificial neural network (ANN). However, CPPNs differ from traditional ANNs in several important ways. Unlike traditional ANNs where every internal node has the same activation function (such as a sigmoid or a step function) CPPN nodes can take on one of several activation functions from a predefined set. This function set often includes functions that are repetitive such as sine or cosine as well as symmetric functions such as Gaussian, thus allowing for motifs seen in natural systems: symmetry, repetition, and repetition with variation. Additionally CPPNs are often used as a generative system to encode some other object of interest e.g. pictures [27], 3D structures [4, 11], robot morphologies [3, 5] or traditional ANNs [31], as opposed to being employed directly as a control architecture as ANNs typically are. Here CPPNs are used as such a generative encoding to produce actuated robot body plans. A more in depth description of CPPNs is beyond the scope of this paper; the reader is referred to [33] for further details.

2.2 Evolutionary Algorithm

In this study CPPN-NEAT [33] is the algorithm used to evolve CPPNs. CPPN-NEAT is an extension of the state of the art NeuroEvolution of Augmenting Topologies (NEAT) [34] method of neuro-evolution. NEAT and by extension CPPN-NEAT begins with small networks (those without any internal or hidden nodes) and gradually increases the complexity of the networks over time through the addition of new nodes and links. Additionally the population is divided into “species” for the purpose of promoting genotypic diversity and allows novel structural innovations time to mature. This systematic way of increasing network complexity as needed should lend itself well to studying how morphologies increase in complexity when evolving inside different environments. A more thorough description of the NEAT and CPPN-NEAT algorithms also falls outside the bounds of this paper, so the reader is directed to [33, 34] for additional details.

2.3 Building Robots from CPPNs

In previous studies [3, 5], robots were constructed out of spherical components from evolving CPPNs by means of an iterated growth procedure. This procedure involved starting at a specific initial point and attaching spheres to grow outwards by means of querying the CPPN genome locally and placing newly created spheres in a priority queue whereby they could be selected as attachment points for additional spheres. This process would repeat until a complete robot was grown.

While promising results were produced by the system presented in those papers it has several drawbacks. In many cases the additional indirection added by the growth procedure prevents desirable features of the CPPNs’ outputs – such as symmetry and repetition – from being realized in the resulting morphologies. Moreover, while spheres are easy to physically simulate due to their single points of contact such that morphologies with a small number of spheres can be cheaply simulated, the computational costs become too large (even on a cluster) when trying to model suffi-

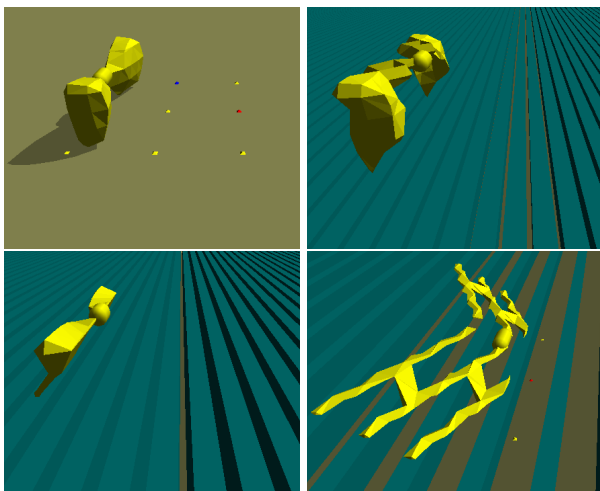


Figure 1: The control environment and a sampling of the experimental environments with robots that evolved to locomote successfully in each. The ground is a high friction surface, while the blue “blocks of ice” have very low friction. To view videos of these robots in action visit <http://tinyurl.com/GECC02012-Videos>

ciently complex physical shapes with spheres. Because of these considerations an alternative method is employed in this work.

In lieu of the growth procedure just described the current study employs a voxel based method to create morphological components out of triangular meshes (trimeshes) similar to what is done for the creation of 3D shapes in [11]. A regular grid is placed over a region of 3D-space which defines the presence of voxel locations. In the current work this region extends from -1 to 1 (inclusive) in each dimension and grid lines are placed at intervals of 0.2 . This yields a total of 11 grid lines in each dimension for a total of 1331 voxels.

A candidate CPPN is iteratively queried with the (x, y, z) Cartesian coordinates at every voxel location except for the extrema in each direction. Voxel locations that exceed a predefined output threshold (0.5 in this case) are considered to contain matter, while those that do not exceed this threshold are considered to be devoid of matter. All voxels lying on one of the extrema ($|x| = 1$ or $|y| = 1$ or $|z| = 1$) are given output value 0 to ensure that the final triangular meshes have completely enclosed surfaces. Once the CPPN has been queried for every voxel location the Marching Cubes algorithm [20] is employed to create triangular meshes from the underlying voxel data. Specifically an enclosed triangular mesh is created for each connected voxel component which defines the exterior surface of a single physical shape. It is these triangular meshes which are sent to the physics simulator where they define the exterior surface of a solid object and are imbued with mass. As far as the authors are aware this is the first instance of physically simulating evolved, rigid body robots composed of triangular meshes.

Since the purpose of this study is to investigate how different task environments affect the shapes of evolved morphologies, a number of simplifications are used in order to concentrate on the physical shapes of the evolved robots and control for other factors that may influence their per-

formance. From the multiple enclosed trimesh components that could be produced when querying a single CPPN only one of these (the largest in terms of number of triangles) is used in the resulting robot. This single component is copied and reflected across the x -axis. The resulting components (the original and its mirror image) are then spread apart by 0.2 units and a capsule of this length is placed between them such that it connects their two closest points. The two trimesh components each connect to this capsule by means of a hinge joint. These joints have rotation normals of $(1, 0, 0)$ and $(0, 0, -1)$ such that the joints rotate through the robot’s coronal and sagittal planes respectively. Reflecting and copying a single component like this ensures that all robots have the same degrees of freedom and ensures that the robots are all bilaterally symmetric (which should make locomotion easier) while at the same time it allows for a vast number of different morphologies due to the flexibility of the trimesh model.

The two degrees of freedom of each robot are actuated by means of coupled oscillators. Each of the two oscillators is parameterized by several parameters: amplitude, period, and phase shift. These six parameters (three parameters apiece for each of the two joints) are directly encoded in the genome of the evolving robots as floating point numbers so that the genome is in actuality a CPPN plus a six dimensional floating point array. These floating point numbers are recombined and mutated in exactly the same manner as CPPN link weights except that since every individual possesses these parameters crossover is possible in all instances of sexual reproduction. Values for these parameters are constrained to predefined ranges: amplitude, $a \in [\frac{\pi}{4}, \frac{3\pi}{4}]$ (so that the hinge rotates between $-a$ and a radians), period $\in [250, 1500]$ simulation time steps (or equivalently $[2, 12]\%$ of the total evaluation time) and phase shift $\in [-1, 1]$ periods. Each parameter has a mutation probability of 0.1 , which was chosen experimentally.

Encoding the control parameters in this fashion is done to keep the controllers as simple as possible so that fitness is primarily dictated by the physical form of the robots while at the same time allowing for diverse enough behavior so that the robots can succeed in the different task environments.

2.4 Selecting desirable robots

The focus of this study is on how varying the complexity of task environments affects the complexity of evolved robot morphologies. Towards this aim a simple task is chosen which can be accomplished with more or less difficulty in a variety of environments. Specifically, like in previous work, the task investigated here is maximizing directed displacement in a fixed amount of time, though this is done across a range of environments and not just on flat ground.

A candidate robot morphology (triangular mesh) and accompanying control parameters are sent to a physics simulator² and allowed to act for a fixed number of simulation time steps. Since trimeshes can be arbitrarily shaped and, unlike spheres, may simultaneously contact the environment at several points it is necessary to use a much smaller step size than has been used in previous work in order to get physically realistic behavior. Specifically, a step size of $0.001s$ is used in this work. Because of this smaller step size a propor-

²Simulations are conducted in the Open Dynamics Engine (<http://www.ode.org>), a widely used open source, physically realistic, simulation environment.

tionally larger number of time steps are needed to achieve the same effective simulation length. Here robots are evaluated for $T = 12500$ time steps.

After the robot has completed its time in the simulator its fitness is calculated. How exactly this fitness is calculated takes some care, because evolution often finds ways to “cheat” naive fitness functions especially when the task environment is difficult. For example, if fitness only considers the positions of the robot’s center of mass, C , and takes fitness as $C(T)_x - C(0)_x$ where $C(t)_x$ is the x -coordinate of the robot’s center of mass at time t and T is the simulation length then in environments where locomotion is difficult evolution will tend to find solutions where C is initially raised far off the ground so that its displacement can be maximized by falling forward. This is a local optimum in this fitness landscape. Similarly, if one tries to eliminate this cheating by only considering the trailing point of the robot so that fitness is $\min p(T)_x - \min p(0)_x$ where $\min p(t)_x$ is the smallest x -coordinate across all points on the robot at time t falling forward can still be an effective solution (and is still a local optimum) in difficult environments if morphologies are created which have backwards protrusions and thus make $\min p(0)_x$ as small as possible.

In light of these considerations the fitness employed in all environments in this research is $\min p(T)_x - \max p(0)_x$. With this fitness function falling forward will not be rewarded because the maximum fitness that can be achieved by pivoting about a single point will be 0 and so a robot must actually displace its whole body forward to be rewarded.

2.5 Exploring environments

As mentioned previously the goal of this study is to investigate how varying the complexity of task environments affects the complexity of evolved robots. To accomplish this goal, robots are evolved in a range of environments with tunable parameters that can effectively increase or decrease the difficulty of the task. For each environment investigated 100 independent evolutionary runs of CPPN-NEAT are run for 500 generations with a population size of 150. The implementation of CPPN-NEAT, the parameter settings, and the CPPN activation functions are the same as those used in [5] except for the addition of the floating point array encoding the control parameters, and an improved selection mechanism³.

The first environment in which robots are evolved in is flat, high friction ground similar to previous work. The robots evolved in this simple environments are considered control cases to compare with robots evolved in other environments. Subsequent environments are more complex: they all consist of an infinite series of low friction rectangular solids (“blocks of ice”) over which a robot must locomote. These “ice blocks” are constructed such that it is impossible for a robot to gain purchase by moving over their upper surfaces but must instead reach into the gaps between the blocks to propel themselves forward. This requires the evolution of morphologies with appropriate physical forms. These “icy” environments vary according to two parameters: the height of the blocks and the spacing between the blocks. Each of

³The authors were made aware through personal correspondence of a bug in the selection mechanism in previous versions of the HyperNEAT C++ distribution. The code was patched to fix this bug (and thus behave as described in the literature) before the current experiments were run.

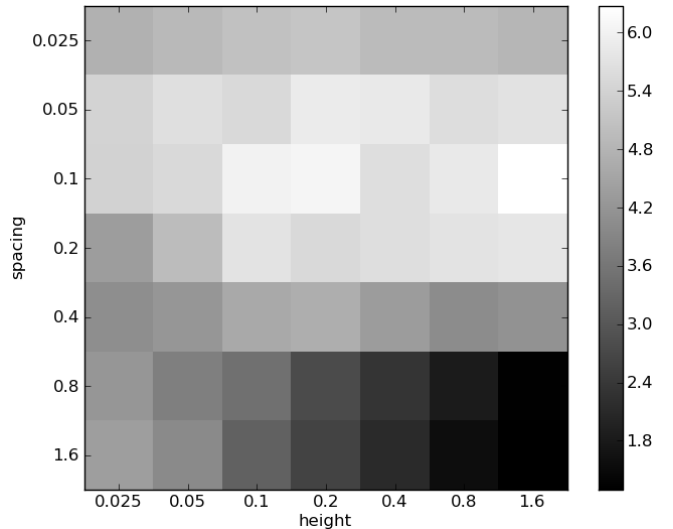


Figure 2: Mean distance achieved (in arbitrary ODE units) by best individual in final generation taken across the 100 independent runs in each of the 49 experimental environments. For comparison the mean distance achieved from the 100 independent runs in the control environment is 5.09 units.

these parameters varies from 0.025 units to 1.6 units exponentially for a total of $7 * 7 = 49$ different environments. The exponential scaling is used in order to cover a range of parameters which produce qualitatively different environments. Figure 1 shows a sampling of these environments and robots that evolve inside them.

3. RESULTS

After completing the 100 run in the control environment and another 100 runs for each of the 49 experimental environments (for a total of $50 * 100 = 5000$ evolutionary runs) the most obvious question becomes: how difficult are these different experimental environments? Or, put another way, how successful is this evolutionary system at producing locomoting robots in each of these environments?

Figure 2 shows the mean distance that the best of run individuals are able to locomote (taken across the 100 independent runs) in each experimental environment. This figure demonstrates that there is a clear relationship between these environmental parameters and the difficulty of the task. Specifically, starting in the lower right of this matrix where both the spacing and the height of blocks are large the task becomes very difficult and the robots all become stuck in the gaps unable to successfully locomote. Keeping the spacing constant and decreasing the block height gradually makes the task easier as the robots are able to navigate over these smaller blocks and therefore displace far enough to be considered successfully locomoting. Once the height has been reduced to 0.025 units the blocks are so small that the environment becomes very similar to flat ground and in fact distances achieved by robots in the lower left environments are not significantly different from those of the control environment, nor are the morphologies in this environment significantly different from those of the control environment (see below).

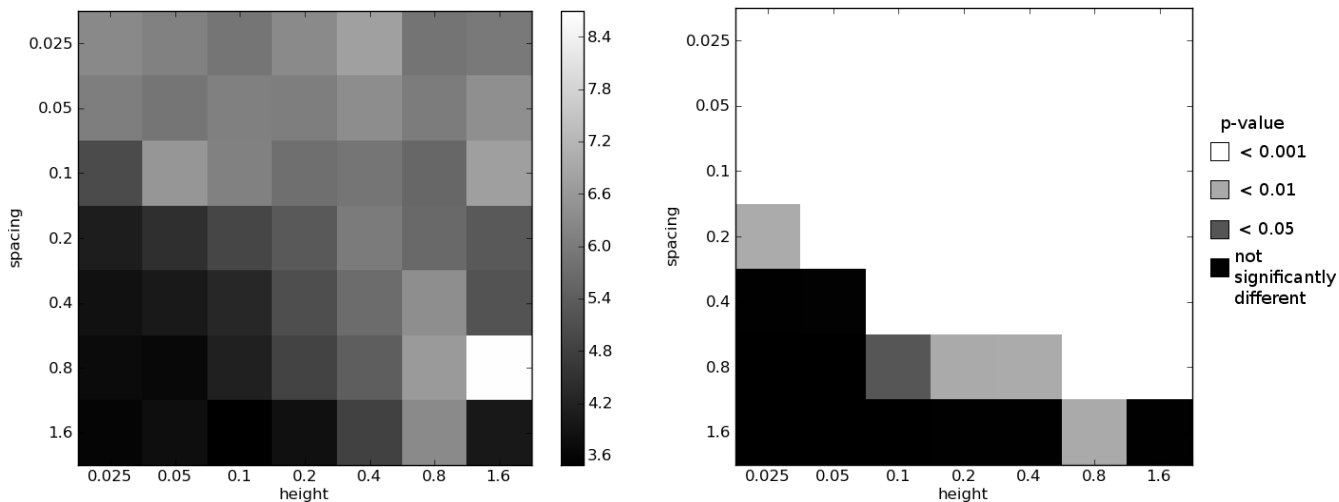


Figure 3: *Left:* Mean ratio between volume of morphology’s AABB and volume of morphology itself for each of the experimental environments. The best of run robots from the control experiment have a mean of 3.58 for this ratio, similar to the black square in this plot. *Right:* Significance of the difference of this ratio in each experimental environment compared to the control environment. The ratio is significantly greater (morphologies are significantly less space filling) on average in the vast majority of experimental environments. There are no experimental environments in which this ratio is significantly smaller than that of the control. All p-values calculated using the Mann-Whitney U test.

As the spacing between the blocks is reduced the robots are no longer able to behave as they would on flat ground, but instead must find ways to move along the tops of the blocks while finding means of gaining purchase by reaching into the gaps. The height of the blocks loses importance in this part of the parameter space but still has an effect (though opposite to when the spacing is large). Here the general pattern is for taller blocks to make the task easier, probably because taller blocks result in a greater volume of space whereby the robot can reach into the gaps to gain purchase. Finally at the top of the matrix, when the spacing is smallest block height ceases to have an impact as no matter what forms the robots evolve to they can not reach very far into the gaps.

For a better understanding of how the evolved robots behave in each of these environments it is helpful to watch them in action. For this purpose, videos of robots evolved in each environment are available on the web at <http://tinyurl.com/GECCO2012-Videos>.

4. DISCUSSION

It is clear that different environments in this parameterization present the evolutionary system with varying degrees of difficulty, but do they also select for different sorts of morphologies? And if so, can these differences be quantified?

One simple way to study this question is to consider how space filling the evolved morphologies are. This can be done by computing the ratio of the volume of a morphology’s Axis Aligned Bounding Box (AABB) to the volume of that morphology itself⁴.

Figure 3 shows the mean values of this ratio, once again taken across the 100 best of run individuals from each experimental environment. Also plotted is how significantly different this ratio is, on average, in each experimental environment when compared to the best of run individuals from the control environment. In the majority of experimental environments this ratio is significantly greater from that of the robots in the control experiment. This demonstrates that these environments do in fact influence the morphologies of the robots which evolve inside them in quantifiable ways: becoming less space filling than those evolved in the control environment for a large portion of the parameter space. Additionally Fig. 3 (left) shows how (at least) one aspect of morphology gradually changes as one moves through this environmental parameter space. This lends support to the chosen parameterization being a good one for the purpose of studying how the morphologies of robots are affected by the environment in which they evolve.

It is clear that the morphologies which evolve in these environments vary in quantifiable (and significant) ways across this parameter space. The question now becomes: do some or all of these environments actually select for more complex morphologies than those that evolve to locomote over flat ground?

There are many ways one might think to quantify the complexity of an evolved morphology. Different measures of how space-filling a morphology is such as the AABB ratio presented above or its surface area to volume ratio or measures of how concave a morphology is (such as the ratio of a morphology’s volume to that of the convex hull of its points) may all hint at how complex a morphology is. However, each of these measures may be deceived by relatively simple body shapes.

⁴For simplicity all morphological measures are computed on the single enclosed trimesh object that is produced by Marching Cubes for a CPPN, i.e. the reflected copy of this trimesh and the connecting capsule are not considered

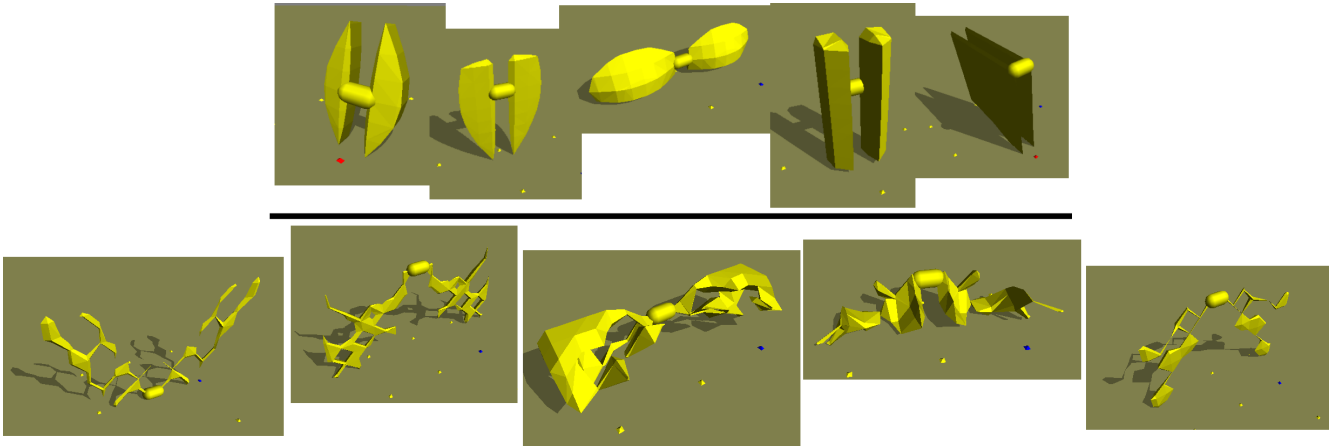


Figure 4: The five morphologies with smallest (*top*) and largest (*bottom*) values of H_{Δ} across all best of run individuals from all environments (experimental and control). The morphologies with high H_{Δ} values are clearly more complex than those with small H_{Δ} values.

4.1 Entropy of curvature

Instead, it is useful to think about the complexity of a body shape in information theoretic terms. One commonly used measure of complexity is Shannon’s Entropy [28], which measures the information content of a random variable. Recent work [24, 35] has demonstrated how notions of Shannon Entropy can be applied to measuring the complexity of a 3D object by considering the curvature of the object as a random variable. In fact, quantifying the complexity of 3D objects in this way has been shown to strongly correlate with human observers notions of complexity [35]. In the space below the building blocks of computing this measure are presented, and the reader is referred to [24] and [35] for more in depth discussions of their theoretical underpinnings.

Given a random variable x with a probability density function (PDF) $p(x)$, entropy H is defined as

$$H = - \int_{-\infty}^{\infty} p(x) \log p(x) dx \quad (1)$$

where $p(x)$ is a continuous function. If instead $p(x)$ is discretized such that $p_i = \int_{x_{i-1}}^{x_i} p(x) dx$ where the x s are specific values of x then H can be formulated discretely as

$$H = - \sum_i p_i \log p_i \quad (2)$$

which is the formulation employed here.

But, what is the random variable x on which H will be calculated? Following [24, 35] x will be a measure of Gaussian curvature of the points on a body shape. Since the body shapes here are built out of triangular meshes the points at which this curvature is non-zero are precisely the vertices of the triangular mesh. Specifically, for each vertex j in a trimesh the angle excess Φ_j is calculated as

$$\Phi_j = 2\pi - \sum_i \phi_i \quad (3)$$

where ϕ_i is the internal angle at j of each triangle i of which j is a vertex. This angle excess Φ_j has a direct relationship to the Gaussian curvature at that point [24]. This will be the variable on which entropy is calculated.

Following the calculation of Φ_j for every vertex a PDF $p(\Phi)$ is estimated by placing the values of Φ_j into discrete bins of uniform width (Δ) and counting the number of Φ_j samples that fall into each bin. This results in a discrete set of probabilities p_i , and Equation 2 can be used to arrive at an estimate of entropy that depends on the chosen Δ , denoted here H_{Δ} ⁵.

Does H_{Δ} calculated in this way capture the complexity of evolved morphologies as has been demonstrated in previous work? To answer this question H_{Δ} is calculated for all 5000 best of run individuals from all environments (experimental and control). Out of those 5000 the five morphologies which have the lowest value for this measure and the five morphologies which have the highest value for this measure are selected. Images of these morphologies are shown in Figure 4. Looking at these two sets of morphologies most everyone would agree that those with high H_{Δ} values appear more complex than those with low H_{Δ} values. In light of this observation and the previous work in this area it is concluded that H_{Δ} does a good job of measuring morphological complexity.

With the knowledge that the complexity of an evolved morphology can be adequately quantified, focus shifts to how the complexity of these morphologies varies from the simple control environment to the more complex parameterized experimental environments. From studying Figure 5 one can see that in total the experimental environments tend to select for more complex morphologies than those which evolve in the control environment. Additionally, there is a suggestive pattern across the parameter space where environments in the upper right half of the matrix are much more

⁵The choice of Δ greatly impacts the results of this calculation. If Δ is too large the majority of samples will fall into the same bin and all information is lost. If Δ is too small then the majority of samples will fall into independent bins and H_{Δ} reduces to a function of the number of vertices n . In general there is no optimum Δ , and since the trimesh morphologies considered here have much fewer vertices than those of [24] a correspondingly larger bin width must be used. In all calculations presented here a bin width $\Delta = \frac{\pi}{10}$ is used, chosen as a reasonable value by visually inspecting histograms of varying bin widths.

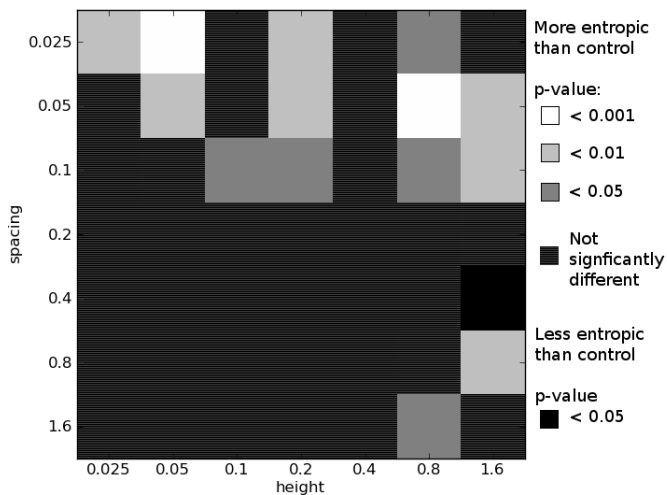


Figure 5: The ways in which morphologies from experimental environments are more, less, or equally complex than those evolved in the control environment. Plot is based on comparing the H_{Δ} values for best of run individuals in each experimental environment to the H_{Δ} values for the best of run individuals in the control environment. The more complex experimental environments tend to select for more complex morphologies: there are many experimental environments where significantly more complex morphologies evolve, while only one experimental environment where significantly less complex morphologies evolve. All p-values calculated using the Mann-Whitney U test.

likely to produce complex morphologies, and this coincides with where the AABB ratios are most significantly different from the control experiment. The upper right of the matrix contains the environments with narrow gaps and higher obstacles, and thus present the most different task from the control environment, so it makes sense that this is where the most different morphologies would evolve. In addition, it makes sense that additional complexity would be needed to succeed in these environments because morphologies with more simple sphere or block like components are unable to reach into the gaps to gain purchase. Therefore evolution must find more complex morphologies to be successful.

5. CONCLUSION

This work has investigated the relationship between environmental and morphological complexity in evolved robots. Using an information theoretic measure of morphological complexity, known to correlate with human perceptions of complexity, it was demonstrated that many complex environments create evolutionary pressures which lead to the evolution of more complex body forms than those of robots evolved in the simple, flat ground environment traditionally investigated. This lends support to the hypothesis that the morphological complexity of a robot should increase commensurately with the complexity of its task environment.

A number of simplifications were made so that the analyses could be focused on the shape of the evolved morphologies. These simplifications included limiting the morpholo-

gies to a specific configuration of connectivity with a single connected trimesh reflected and copied and connected via hinge joints to an intermediary capsule, and using an open loop control strategy of coupled oscillators with a small number of evolvable parameters. While the robots evolved in this manner were able to successfully locomote in the majority of environments investigated it would be interesting to investigate how removing these simplifications would affect the presented results. Perhaps with a more sophisticated controller and/or a greater number of degrees of freedom it would be possible to evolve robots which succeed in the most challenging environments or that are able to succeed in other environments without such an increase in morphological complexity: could increased control complexity supplant the need for increased morphological complexity? However, quantifying the complexity of such robots will require extending the entropy measure presented here to take into account additional factors such as the number and placement of additional degrees of freedom and the complexity of their controller architectures.

The estimation of probability density functions used here in calculating H_{Δ} could also be improved. As mentioned above, the chosen method depends heavily on selecting a good bin width Δ . Alternative means of estimating a PDF such as kernel density estimation [29] may provide a better means of calculating this measure and will be investigated in future research.

Finally, it will be interesting to see how the morphologies of evolved robots vary in other environments not experimented with in this work. Do certain environments drive an increase in morphological complexity while others drive an increase in complexity of the control strategy or sensory system? Could measuring the complexity of robots while they are evolving be used to inform the evolutionary search process in a meaningful way? Could environments co-evolve along with morphologies much like natural environments change over time, and therefore implicitly drive the evolution of complexity in a more principled way? All of these are fruitful areas for future research.

6. ACKNOWLEDGMENTS

This work was supported by National Science Foundation Grant PECASE-0953837 and DARPA M3 grant W911NF-1-11-0076.

The authors also acknowledge the Vermont Advanced Computing Core which is supported by NASA (NNX 06AC88G), at the University of Vermont for providing High Performance Computing resources that have contributed to the research results reported within this paper.

7. REFERENCES

- [1] A. Adamatzky, M. Komosinski, and S. Ulatowski. Software review: Framsticks. *Kybernetes: The International Journal of Systems & Cybernetics*, 29(9/10):1344–1351, 2000.
- [2] M. Anderson. Embodied Cognition: A field guide. *Artificial Intelligence*, 149(1):91–130, 2003.
- [3] J. E. Auerbach and J. C. Bongard. Dynamic Resolution in the Co-Evolution of Morphology and Control. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*, 2010.

- [4] J. E. Auerbach and J. C. Bongard. Evolving CPPNs to Grow Three-Dimensional Physical Structures. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2010.
- [5] J. E. Auerbach and J. C. Bongard. Evolving Complete Robots with CPPN-NEAT: The Utility of Recurrent Connections. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2011.
- [6] R. D. Beer. The dynamics of brain-body-environment systems: A status report. In P. Calvo and A. Gomila, editors, *Handbook of Cognitive Science: An Embodied Approach*, pages 99–120. Elsevier, 2008.
- [7] J. Bongard and R. Pfeifer. Repeated structure and dissociation of genotypic and phenotypic complexity in Artificial Ontogeny. *Proceedings of The Genetic and Evolutionary Computation Conference (GECCO 2001)*, pages 829–836, 2001.
- [8] J. C. Bongard. Evolving modular genetic regulatory networks. In *Proceedings of The IEEE 2002 Congress on Evolutionary Computation (CEC2002)*, pages 1872–1877, 2002.
- [9] R. Brooks. *Cambrian intelligence*. MIT Press Cambridge, Mass, 1999.
- [10] J. Clune, B. Beckmann, C. Ofria, and R. Pennock. Evolving Coordinated Quadruped Gaits with the HyperNEAT Generative Encoding. In *Proceedings of the IEEE Congress on Evolutionary Computing*, pages 2764–2771, 2009.
- [11] J. Clune and H. Lipson. Evolving 3d objects with a generative encoding inspired by developmental biology. In *Proceedings of the Eleventh European Conference on Artificial Life (ECAL)*, pages 144–148, 2011.
- [12] J. Clune, R. T. Pennock, and C. Ofria. The sensitivity of hyperneat to different geometric representations of a problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2009.
- [13] P. Eggenberger. Evolving morphologies of simulated 3D organisms based on differential gene expression. *Procs. of the Fourth European Conf. on Artificial Life*, pages 205–213, 1997.
- [14] I. Harvey, P. Husbands, D. Cliff, A. Thompson, and N. Jakobi. Evolutionary robotics: the sussex approach. *Robotics and Autonomous Systems*, 20:205–224, 1997.
- [15] G. Hornby and J. Pollack. Body-brain co-evolution using l-systems as a generative encoding. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 868–875, 2001.
- [16] G. Hornby and J. Pollack. Evolving L-systems to generate virtual creatures. *Computers & Graphics*, 25(6):1041–1048, 2001.
- [17] M. Komosinski and A. Rotaru-Varga. Comparison of different genotype encodings for simulated three-dimensional agents. *Artif. Life*, 7(4):395–418, 2002.
- [18] N. Lassabe, H. Luga, and Y. Duthen. A new step for artificial creatures. In *Proceedings of 1st IEEE Conference on Artificial Life (IEEE-ALife 2007)*, pages 243–249. IEEE Press, 2007.
- [19] H. Lipson and J. B. Pollack. Automatic design and manufacture of artificial lifeforms. *Nature*, 406:974–978, 2000.
- [20] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21:163–169, August 1987.
- [21] H. H. Lund and J. W. P. Lee. Evolving robot morphology. *IEEE International Conference on Evolutionary Computation*, pages 197–202, 1997.
- [22] C. Mautner and R. Belew. Evolving robot morphology and control. *Artificial Life and Robotics*, 4(3):130–136, 2000.
- [23] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology*. MIT Press, Cambridge, MA, USA, 2000.
- [24] D. Page, A. Koschan, S. Sukumar, B. Roui-Abidi, and M. Abidi. Shape analysis algorithm based on information theory. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I – 229–32 vol.1, sept. 2003.
- [25] C. Paul. Morphological computation: A basis for the analysis of morphology and control requirements. *Robotics and Autonomous Systems*, 54(8):619–630, 2006.
- [26] R. Pfeifer and J. Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, 2006.
- [27] J. Secretan, N. Beato, D. B. D’Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-Kovarik, and K. O. Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation Journal*, pages 373–403, 2011.
- [28] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.
- [29] B. W. Silverman. *Density estimation: for statistics and data analysis*. London, 1986.
- [30] K. Sims. Evolving 3D morphology and behaviour by competition. *Artificial Life IV*, pages 28–39, 1994.
- [31] K. Stanley, D. D’Ambrosio, and J. Gauci. A Hypercube-Based encoding for evolving Large-Scale neural networks. *Artificial Life*, 15(2):185–212, 2009.
- [32] K. Stanley and R. Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, 2003.
- [33] K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162, 2007.
- [34] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:2002, 2001.
- [35] S. Sukumar, D. Page, A. Koschan, and M. Abidi. Towards understanding what makes 3d objects appear simple or complex. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008, Sixth IEEE Workshop on Perceptual Organization in Computer Vision (POCV)*, 2008.