Evolving Three-Dimensional Objects with a Generative Encoding Inspired by Developmental Biology

Jeff Clune and Hod Lipson

Department of Mechanical and Aerospace Engineering, Cornell University
jeffclune@cornell.edu

Abstract

This paper introduces an algorithm for evolving 3D objects with a generative encoding that abstracts how biological morphologies are produced. Evolving interesting 3D objects is useful in many disciplines, including artistic design (e.g. sculpture), engineering (e.g. robotics, architecture, or product design), and biology (e.g. for investigating morphological evolution). A critical element in evolving 3D objects is the representation, which strongly influences the types of objects produced. In 2007 a representation was introduced called Compositional Pattern Producing Networks (CPPN), which abstracts how natural phenotypes are generated. To date, however, the ability of CPPNs to create 3D objects has barely been explored. Here we present a new way to create 3D objects with CPPNs. Experiments with both interactive and target-based evolution demonstrate that CPPNs show potential in generating interesting, complex, 3D objects. We further show that changing the information provided to CPPNs and the functions allowed in their genomes biases the types of objects produced. Finally, we validate that the objects transfer well from simulation to the real-world by printing them with a 3D printer. Overall, this paper shows that evolving objects with encodings based on concepts from biological development can be a powerful way to evolve complex, interesting objects, which should be of use in fields as diverse as art, engineering, and biology.

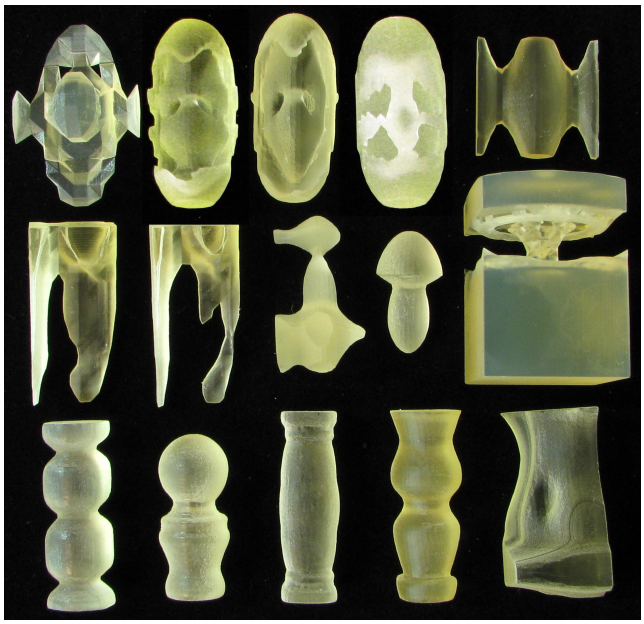


Figure 1: Examples of evolved objects that were transferred to reality via a 3D printer.

Motivation and Previous Work

The diversity, complexity, and function of natural morphologies is awe-inspiring. Evolution has created bodies that can fly, run, and swim with amazing agility. It would be desirable to harness the power of evolution to create synthetic physical designs and morphologies. Doing so would benefit a variety of fields. For example, artists, architects and engineers could evolve sculptures, buildings, product designs, and sophisticated robots. Evolution should be especially helpful in the design of complex objects with many interacting parts made of non-linear materials. In such challenging problem domains, evolution excels while human intuition is limited. Being able to evolve sophisticated morphologies also furthers biological research because it enables the investigation of how and why certain natural designs were produced. Evolving 3D objects is thus worthwhile both as a basic science and for its innumerable potential applications. This paper describes how 3D shapes can be evolved and then transferred to reality via 3D printing technology (Figure 1).

Previous research in digital morphological evolution has typically involved encodings that were either highly biologically detailed, or highly-abstract with less biological accuracy. The former camp frequently simulates the low-level processes that govern biological development, such as the diffusing *morphogen* chemicals and proteins that determine the identity of embryonic cells (Bongard and Pfeifer 2001, Eggenberger 1997, Miller 2004). While this approach facilitates studying the mechanisms of developmental biology, the computational cost of simulating chemistry in such detail greatly limits the complexity of the evolved phenotypes. The most complex forms typically evolved in such systems are simple geometric patterns (such as three bands) (Miller

2004) or groups of shapes resembling the earliest stages of animal development (Eggenberger 1997).

The second camp employs high-level abstractions that enable the evolution of more elaborate forms with many parts, but these abstractions tend not to reflect the way that organisms actually develop (Wolpert and Tickle 2010, Bentley 1996). An example is Lindenmayer Systems (L-Systems), which iteratively replace symbols in strings with other symbols until a termination criteria is reached (Lindenmayer 1968, Hornby et al. 2003). While L-Systems can reproduce a wide variety of organismal shapes, especially those of branching plants, they do not model plant developmental processes (Wolpert and Tickle 2010). Another example is the work of Sims (1994), who evolved morphologies that resembled some biological creatures, although with an abstract encoding based on parameterized recursion that does not resemble natural developmental processes (Sims 1994).

A third option is possible, wherein a high-level abstraction is based on the developmental processes that give rise to natural forms. An example of this approach is Compositional Pattern Producing Networks (CPPNs) (Stanley 2007), which are used to evolve 3D objects in this paper and are described in Methods. Two groups have previously evolved 3D objects with CPPNs, although neither conducted an open-ended exploration of 3D objects. One group evolved CPPN objects that were composed of variable-sized spheres and were evaluated on two tasks: falling (Auerbach and Bongard 2010b) or moving rapidly (Auerbach and Bongard 2010a). Most of the evolved forms resembled clubs. A second group evolved soft-bodied robots to move quickly (Hiller and Lipson 2010). These studies demonstrate that CPPNs can create functional shapes, but leave open the question of what types of 3D objects CPPNs can produce with fewer constraints and without specific objectives.

2D pictures are evolved with CPPNs on picbreeder.org, where humans perform selection (Secretan et al. 2011). The complexity and natural appearance of the resulting images often support claims regarding the legitimacy of CPPNs as an abstraction of biological development (Stanley 2007). A demonstration in 3D would significantly strengthen these claims, however, because the natural world is 3D. It is possible that CPPNs are unable to frequently make sensible forms with the added difficulty of another dimension, and when objects must be one contiguous unit (which aids in transfers to reality). A recent paper by Bánsági Jr et al. (*Science* 2011) highlights the need to verify that generative encodings that produce complex patterns in 2D also can do so in 3D. By evolving CPPN objects in the natural 3D setting, this paper conducts a critical test of the hypothesis that generative encodings based on *geometric* abstractions of development capture some of the complexity-generating power of natural morphological development. Doing so also provides a visually intuitive testbed for studying how variants of such generative encodings behave. It also reveals the utility of CPPNs as a representation for 3D object design.

Methods

Compositional Pattern Producing Networks

Compositional Pattern Producing Networks (CPPNs) abstract the process of natural development without simulating the low-level chemical dynamics involved in developmental biology (Stanley 2007). Cells (and higher-level modules) in natural organisms often differentiate into their possible types (e.g. heart or spleen) as a function of where they are situated in geometric space (Wolpert and Tickle 2010).

Components of natural organisms cannot directly determine their geometric location, so developmental processes have evolved to create gradients of chemicals and proteins called morphogens that organismal components use to figure out *where* they are and, thus, *what* to become (Wolpert and Tickle 2010). For example, in many animals the anterior-posterior and dorsal-ventral axes are specified by maternally provided morphogen gradients. Embryonic genes then construct more complicated geometric patterns of morphogens as a function of these simpler gradients. Downstream genes can construct additional pattern as a function of any of the patterns already created, enabling the production of patterns of arbitrary complexity (Wolpert and Tickle 2010).

CPPNs abstract this process by allowing similar geometric patterns to be composed of other geometric patterns, but represent the patterns mathematically instead of via diffusing morphogens. To replace maternally-provided gradients, the experimenter provides the initial gradients. Final patterns output by the CPPN determine the attributes of the phenotypic components at different geometric locations. For example, two-dimensional pictures could be encoded by iteratively passing the coordinates of each pixel on a canvas (e.g. $x = 2, y = 4$) to a CPPN genome and having the output specify the color or shade of each pixel (Figure 2).

Each CPPN is a directed graph in which every node is itself a single function, such as sine or Gaussian. The nature of the functions can create a wide variety of desirable properties, such as symmetry (e.g. a Gaussian function) and repetition (e.g. a sine function) that evolution can exploit. Because the genome allows functions to be made of other functions, coordinate frames can be combined. For instance, a sine function early in the network can create a repeating theme that, when passed into the symmetrical Gaussian function, creates a repeating series of symmetrical motifs (Figure 2). This process abstracts the natural developmental processes described above (Wolpert and Tickle 2010).

The links that connect and allow information to flow between nodes in a CPPN have a weight that can magnify or diminish the values that pass along them. Mutations that change these weights may, for example, give a stronger influence to a symmetry-generating part of a network while diminishing the contribution from another part.

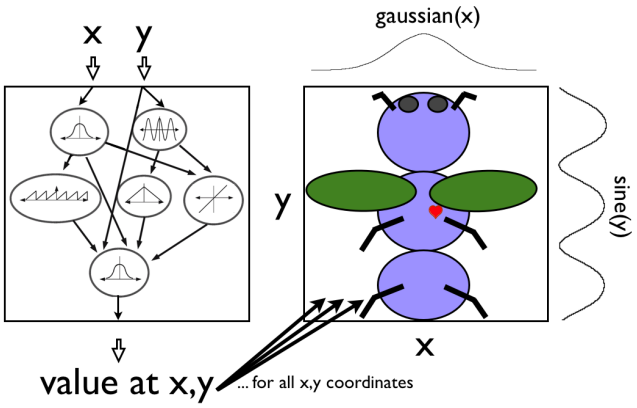


Figure 2: CPPNs combine mathematical functions to create regularities, such as symmetries and repeated modules, with and without variation. Adapted from Stanley (2007).

Variation is produced by mutating or crossing CPPNs. Mutations can add a node or change weights. The default set of allowable functions for CPPNs in this paper are sine, sigmoid, Gaussian, and linear, although we also experimented with additional functions (see Results). The evolution of the population of CPPN networks occurs according to the principles of the NeuroEvolution of Augmenting Topologies (NEAT) algorithm (Stanley and Miikkulainen 2002).

The NEAT algorithm contains three major components (Stanley and Miikkulainen 2002). (1) It starts with small genomes that encode simple networks and complexifies them via mutations that add nodes and links to the network. This complexification enables the algorithm to evolve the network topology in addition to its weights. (2) NEAT preserves diversity via a fitness-sharing mechanism that allows new innovations time to be tuned by evolution before competing them against more optimized rivals. (3) crossover utilizes historical information in a way that is effective, yet avoids the need for expensive topological analysis.

Encoding 3D Objects with CPPNs

To evolve 3D objects, inputs for the x, y, and z dimensions are provided to a CPPN. Additional gradients can be provided, which may bias the types of objects produced (see Results). A *workspace* (maximum object size) is defined with a *resolution*, which determines the number of voxels in each dimension. In this paper there are 10 voxels in the x and z dimensions and 20 in the y (vertical) dimension. The x, y, and z value of each voxel are iteratively input to a CPPN, and voxels are considered full if the CPPN output is greater than a threshold (here set to 0.1), otherwise the voxel is considered empty. The 3D voxel array is then processed by the surface-smoothing Marching Cubes algorithm (Lorensen and Cline 1987). A normal is provided for each vertex when visualizing the objects in OpenGL, a graphics technique that further smooths the surface. These two smoothing steps enable high-resolution CPPN objects to be visualized without prohibitive computational costs.

This algorithm for encoding 3D objects is a more straightforward extension of how CPPNs encode 2D pictures (Stanley 2007, Secretan et al. 2011) than another algorithm for evolving 3D objects with CPPNs, which included growth over time and limited shapes to collections of attached spheres of different sizes (Auerbach and Bongard 2010b;a).

Selection Mechanisms (Fitness Assignment)

We evolve images with interactive evolution and target-based evolution. During interactive evolution the user (here, the first author) views N rotating objects (here, 15) and selects a champion, which receives a fitness of 1000. The user can also reward additional organisms that receive a fitness of 500. To avoid uninteresting objects, those that are not chosen, yet have voxel counts between 10% and 90% of the maximum number possible, are given a fitness of 100. The remaining objects are given a fitness of 1. For target evolution, the fitness is the percent of voxels that matched the target object. To magnify differences in fitness values, all fitness scores serve as an exponent to a large constant $c = 2000$ to produce the final fitness value. The parameters are identical to a previous work (Clune et al. 2011), except mutations were allowed to be larger (MutationPower = 2.5).

Results and Discussion

Interactive Evolution

Overall summary We study interactive evolution because it allows an open-ended exploration of the design space of objects CPPNs can produce. Additionally, interactive evolution avoids the greedy nature of target-based evolution, potentially allowing it to access more interesting objects (Secretan et al. 2011, Lehman and Stanley 2008). A drawback of interactive evolution is that it is subjective, but science should not abandon such a useful tool simply because it is subjective. While user preferences bias the types of objects selected, the encoding has to be able to produce such objects in the first place in order for them to be selected. Different encodings will bias the types of patterns evolved (Clune et al. 2011), meaning that interactive evolution can inform us about the biases and expressive power of the encoding.

Figure 3 shows example objects from different generations during a run of interactive evolution. The geometric patterns become more complex over generations, which reflects the property of complexification built into NEAT (Stanley and Miikkulainen 2002).

Figure 4 displays a few of the interesting objects discovered in different runs, some of which had different inputs and parameters (described below). It is important to note that these objects were chosen from a small number of runs performed by one person, each of which was limited to tens or perhaps a few hundred generations. It is noteworthy that such recognizable 3D forms emerge in such a

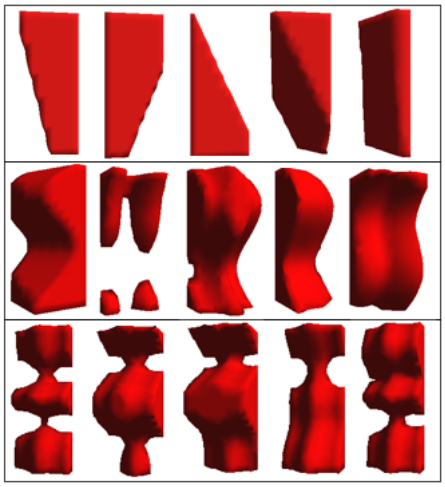


Figure 3: Representative objects from different generations of a single run of interactive evolution. From top to bottom, rows display individuals from generations 1, 15, and 33.

small sample size. These 3D objects should not be held to the same standard as pictures from picbreeder.org, where hundreds of users have published thousands of images after performing over 150,000 evaluations across hundreds of generations (Secretan et al. 2011).

The objects in Figure 4 exhibit many properties that are desirable both for studying morphological evolution and harnessing it for engineering or artistic purposes. The objects are frequently regular, a property which is important in engineering and for evolvability (Lipson 2007, Clune et al. 2011). An important regularity is symmetry, which is evident with respect to different dimensions in many of the objects. For example, all of the objects in generation 33 of Figure 3 are highly left-right symmetric, and objects b7 and b8 in Figure 4 exhibit left-right and top-bottom symmetries. Another useful regularity is repetition, which occurs frequently in the evolved objects (e.g. the top-right object in Figure 3). A further beneficial property is exhibiting regularity with variation (Stanley and Miikkulainen 2003, Lipson 2007, Clune et al. 2011). For example, Figure 4b1 has a motif that appears like an animal head, but is repeated in different sizes and with other subtle variations. Symmetric patterns with asymmetric variations can also be observed, such as in Figure 4a8 and Figure 4b6.

It is important to note that humans often select regular, symmetrical shapes, which increases their frequency in interactive evolution. That said, biology and engineering also often reward regularity. Additionally, it has been shown that when CPPNs generate artificial neural networks that control robots in target-based evolution, the neural wiring patterns are often regular, including symmetries and repeated themes (Clune et al. 2011), demonstrating that CPPNs produce regularities even without humans performing selection.

Most importantly, the evolved objects often look similar to natural forms or engineered designs, revealing that CPPNs can produce the types of objects we are interested in designing and studying with synthetic morphological evolution. Humans can only select such such familiar forms if an encoding tends to produce such designs, which has not been the case for most previous generative encodings. People often describe Figure 4a2 and 4a3 as faces, 4a4 as a Jack-o'-lantern face, 4a5 as an animal figurine, 4a6 as an African statue of a human, 4a7 as a human female stomach, 4a8 as a human female torso, 4b1 and 4b4 as animals, 4b2 and 4b3 as elephants, 4b5 as a human head and shoulders, 4b6 as a horned mask, and 4b7 and 4b8 as spaceships. Some also describe 4b7 as a butterfly. People describe other objects as interesting art, even though they do not resemble any specific natural or human design (e.g. Figure 4a1). Such objects can potentially spark artistic ideas for new forms. The fact that the shapes consistently evoke human and natural designs demonstrates the expressive power of the CPPN encoding to produce interesting 3D objects.

An additional important property is that the offspring of the 3D CPPN objects are similar to their parents, but are varied in interesting ways. Some encodings lack this property in that mutations have dramatic effects, rendering most offspring very different from their parents, which hinders evolvability (Stanley and Miikkulainen 2003). For example, Figure 4b4 is the child of Figure 4b3, and Figure 4b2 is their close relative. All three are consistently described as animals, yet are interesting variations on the animal theme. For example, only a single generation of genetic changes between Figure 4b3 and Figure 4b4 transformed what appears like an elephant with a trunk into something resembling an elephant with warthog tusks. A different variant of Figure 4b3 that thickened the trunk can be seen in Figure 1 (center row, left), which is next to a printed copy of Figure 4b3. Moreover, Figure 4b3, its relative in Figure 1, and Figure 4b2 all evoke elephants, but they are quite different objects, suggesting that the CPPN has captured some fundamental aspects of the elephant concept that it expresses in different ways.

Some of the geometric complexity in the genome is not visible in these 3D phenotypes because a threshold determines the presence or absence of a voxel. In contrast, picbreeder pictures have a continuum of outputs in grayscale and color, which adds to their complexity. Pre-thresholded geometric information could be useful, however, to make colored 3D objects, or to have objects with multiple materials (e.g. the soft-robot equivalent of muscle and bone).

Varying CPPN parameters generates different objects To test whether the types of objects produced could be biased by the CPPN inputs and parameters, we performed multiple runs of interactive evolution with varying conditions. We initially provided only x, y, and z values for

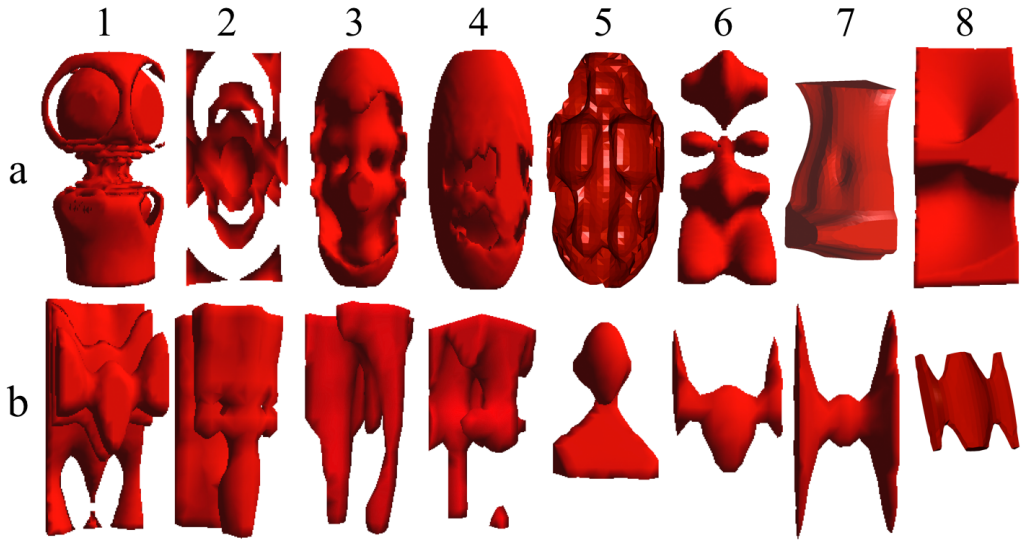


Figure 4: Example objects evolved with CPPNs via interactive evolution.

each voxel. Even with this minimal information, regularities such as symmetries and repeating themes were common (Figure 3), which is expected in a generative encoding with symmetric and repeating genomic functions. The objects in this setup seemed to require more generations before they became interesting, and usually did not appear like objects floating in space, but instead bordered the workspace wall.

We then added the distance from center as an input to the CPPN, which picbreeder also has (in 2D) (Secretan et al. 2011). This information more frequently created rounded objects centered in space. Because the distance-from-center function took the normalized values in each dimension, and the y (height) dimension was longer, an egg-shaped motif was common (Figure 5, left three). All of the objects in Figure 4 have this input. Preliminary experiments with other inputs also revealed interesting biases in the resulting objects (not shown), suggesting a rich area of research regarding how best to bias CPPNs with seed gradients.

To date, no published results explore how patterns differ when recurrence is allowed in CPPN genomes. We enabled recurrence and discovered that the resulting patterns are qualitatively different in that they tend to include fractal patterns. For example, branching patterns emerged, such as an object resembling a tree (Figure 6, left) and another evoking the vascular system (Figure 6, center). Like with fractals, the complexity is often concentrated at the surface boundary, producing a jagged surface effect (e.g. Figure 6, right). Objects with recurrent genomes were much more likely to have small, separated pieces floating in space.

Another interesting parameter of CPPNs is the set of possible genomic node functions. No research published to date has tested different function sets on the same problem to understand how CPPN patterns are affected by this parameter. Visual domains such as 3D objects are a helpful place to start such explorations because of the intuition they provide. We added a square, cosine, and sign-preserving square root function and performed additional runs. Objects in these runs tend to be more complex in earlier generations, and seem to involve both rounded and sharp edges. Figure 4b7 and the rightmost three in Figure 5 are example objects evolved with this expanded genomic node function set.

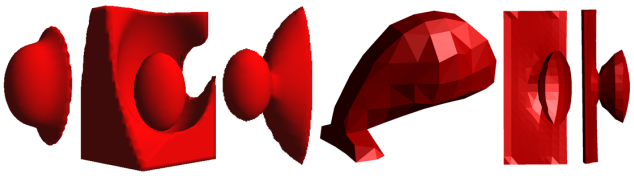


Figure 5: Objects evolved with a distance-from-center input (left three), which frequently featured egg-shape motifs, and objects evolved with an expanded set of genome functions (right three). The rightmost two images show different angles of the same object. Facets in the right three objects result from a close zoom and because, for illustration, normals are provided for facets instead of vertices.

Target-based Evolution

A second way to explore the capabilities of CPPNs is to challenge them to produce a target object. Knowing how CPPNs perform in 3D in target-based evolution is helpful for numerous reasons. Initially, it serves as a preliminary test of how CPPNs might perform on more open-ended, yet still target-based problems, such as evolving robot morphologies to perform certain tasks (e.g. locomotion). Additionally, biologists would benefit if they could repeatedly evolve var-

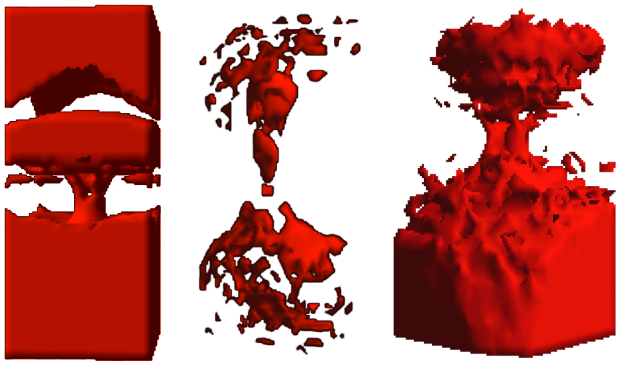


Figure 6: Example objects with recurrent genomes.

ious morphologies to study whether certain developmental strategies for constructing 3D geometric patterns arise frequently. Finally, target evolution allows an artist or engineer to explore objects that are similar to a target object, yet differ in interesting ways (similar to how Figure 4b4 and Figure 4b2 result from slight permutations to the genome of Figure 4b3). Finally, target-based evolution is much faster, enabling an exploration the effects of different parameter settings, which can inform interactive evolution.

The target object for this paper is shown in Figure 8a. It consists of four partially-overlapping spheres, with the outer two halved by workspace bounding box. This target has round shapes that are different from the egg-shaped motif facilitated by the distance-from-center input, providing a test of whether such a related input improves performance. Each treatment has 20 runs with a population of 150 for 1000 generations, unless otherwise specified.

The baseline treatment featured only x, y, and z inputs and the default set of genome functions. The best performing object in each run captures the long cylindrical shape of the target, but most attempts at rounded edges are imperfect combinations of straight-line functions. All runs except one failed to carve much material away between the spheres. An average of 90.8% ($\pm$ 0.003 SE) of voxels are matched (Figure 7), but the target object is not identifiable until about $\geq$ 93% of voxels are matched. As such, the small differences in fitness between the treatments in Figure 7 represent substantial differences in whether the target object is recognizable. Interestingly, one outlier run in this treatment performed much better than the rest (with 94.6% of voxels correct). It features rectangular approximations of spheres (Figure 8b). The lack of round shapes in this treatment corroborates the previous subjective observation from interactive evolution that CPPNs can struggle to evolve and exploit round gradients when they are not provided as inputs.

To test if seeding CPPNs with spherical gradients makes it easier to match this rounded target, we added distance to the center as an input. The CPPNs in the previous treatment could have evolved to calculate this same information, but that may have been difficult. Surprisingly, this information significantly lowered performance to 90.0% ($\pm$ 0.002 SE, $p = 0.013$, Mann-Whitney test, Figure 7). However, the evolved objects all have smooth, round forms (Figure 8c-d), confirming that providing different seed gradients can bias the types of evolved objects. While this might be expected in early generations, it is interesting that the gradients provided have noticeable effects after a thousand generations. This result is in line with a previous paper that found that the information input into CPPNs can bias the resulting phenotypes (Clune et al. 2009). We include this input in the remaining treatments in this paper because it facilitates round surfaces, even though it hurt performance in this experiment.

Because interactive evolution features smaller population sizes, it is worthwhile to study how this difference affects the search for 3D objects. Additionally, since NEAT complexifies genomes over evolutionary time, having more generations may improve the search by accessing genomes with more hidden nodes. We investigate these issues by decreasing the population size from 150 to 15 and increasing the number of generations tenfold to 10^4, which keeps the number of evaluated objects the same. This change significantly improves performance to 91.8% ($\pm$ 0.003 SE, $p < 0.001$, Mann-Whitney test, Figure 7), suggesting that the small population sizes in interactive evolution do not hurt, and may actually benefit, morphological evolution with NEAT-based encodings. The evolved objects tend to have more space carved out between the spheres (Figure 8e-f).

A fundamental evolutionary parameter that can greatly affect evolvability is the mutation rate. We varied the major sources of mutation in NEAT by altering the rate at which genomic links are added, removed, and mutated, as well as the rate at which genomic nodes are added. Increasing the node addition rate significantly boosted performance ($p < 0.001$, Mann-Whitney test, Figure 7) to 91.5% ($\pm$ 0.003 SE). Changing the other mutation rate parameters did

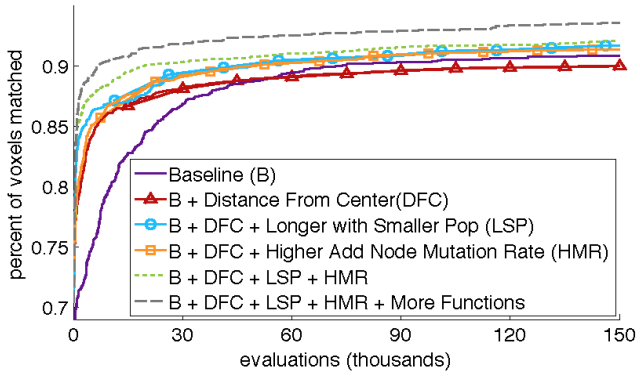


Figure 7: Means of the best-performing individuals for target-based evolution. See text for variance.

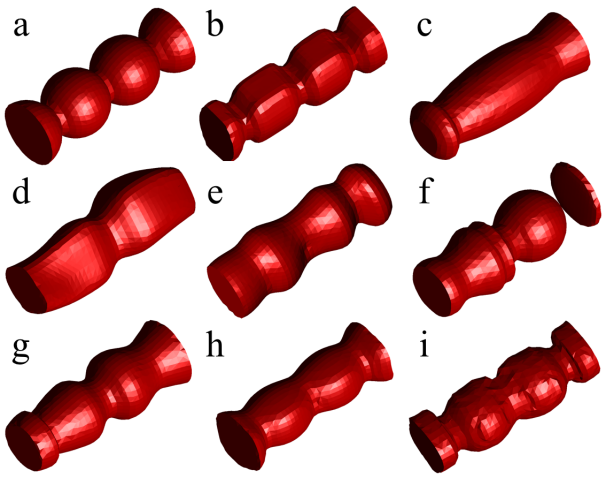


Figure 8: Target-based evolution objects.

not improve performance (data not shown).

Because a smaller population with more generations was beneficial, and because a higher mutation rate was beneficial, we tested whether both changes together would outperform either alone. The combination did improve performance to 92.0% (Figure 7), but the difference was not significant ($p > 0.05$, Mann-Whitney test). We also found that the expanded genome function set (described previously) improved performed to 93.0%, which was significant ($p = 0.022$, Mann-Whitney test). As before, the objects in this treatment seemed to combine rounded surfaces with sharper edges: while most were smooth (e.g. Figure 8g-h), a few had rough patches on their surface, including Figure 8i. Adding recurrent genomic connections to this treatment did not significantly affect performance (93.3%, $p > 0.05$).

Overall, the target-based evolution experiments reveal that evolving CPPNs can roughy match a target object. While a high percentage of voxels were matched, the degree to which the evolved objects qualitatively resemble the target is subjective and debatable. The most important contribution of these experiments is to better understand the way in which target-based evolution is biased by different parameters. These results are preliminary, however, until more tests can be conducted with additional targets.

It is also interesting that many of the evolved objects look designed for a purpose. For example, many of the objects in Figure 8 seem like functional and aesthetically attractive objects carved on a lathe, such as legs from tables and chairs or posts from banisters and railings. One reason this is surprising is because it could have been the case that the greedy nature of target-based evolution would have gained improvements by iteratively adding small patches of voxels that match a subset of the overall space. Such a patchwork solution would not look as regular and smooth as the objects that actually evolved, suggesting that CPPNs are biased away from such a piecemeal strategy. Previous work has shown that CPPNs have difficulty making exceptions to regular patterns when evolving neural networks (Clune et al. 2011), which could explain why the target object in this study was not matched one patch at a time. Such a bias toward regularity may simultaneously explain the smoothness of the evolved objects and why matching the final few percent of voxels is so difficult.

Artists and engineers may actually benefit from the fact that the evolved objects share some properties of the target, but are different in interesting ways. This means that a designer can provide a seed object as a target, and a series of objects can automatically be generated that are aesthetically interesting variations on that seed concept (Figure 8).

Transferring Objects to the Physical World

Advances in 3D printing technologies make it possible to transfer evolved objects into the physical world, which may help artists and engineers benefit from this technology. To test whether CPPN objects maintained their appearance and structural integrity in reality we printed them on a Connex500 3D printer. The objects look similar to their simulated counterparts and are structurally sound (Figure 1). One difference is that non-contiguous pieces (e.g. the top of Figure6, left) are not held in place in the physical world without additional scaffolding. By printing in a semi-transparent material, we also discovered that none of the objects have visible hollow areas embedded within them, although CPPNs can create such negative spaces. While the gap between simulated and physical objects was not expected to be large for static objects, it is helpful to have verified the fidelity of the transfer.

Conclusions and Future Work

This paper introduces an algorithm for evolving 3D objects with the CPPN generative encoding, which is a computationally efficient abstraction of biological development. We conducted both interactive and target-based evolution to explore the ability of CPPNs to create complex objects, especially those that resemble natural and engineered designs.

A small, preliminary exploration of the design space of 3D CPPN objects unearthed a diversity of objects that evoke natural and engineered forms. Many of the objects featured regularities such as symmetry and repetition, with and without variation. Such properties are important for engineering and evolvability (Lipson 2007, Clune et al. 2011), and suggest that CPPNs are a promising encoding for evolving useful and aesthetically pleasing objects. To extend this research we are creating a website like picbreeder.org (Secretan et al. 2011) where users can collaboratively evolve 3D objects online, which will provide a much larger exploration of the potential of this technology. It will also overcome the need for any individual to perform all of the evaluations in a lineage and thus allow more complex objects to evolve.

Experiments with target-based evolution on one target revealed how the inputs and parameters of CPPNs can influence the types of objects they evolve. The evolved objects roughly resemble the target, but do not match it precisely. While the evolved objects share some properties of the target, they also differ from it in interesting ways. This property could help artists and engineers by providing 3D designs that are variations on a seed concept. All of these conclusions are tentative, however, since experiments were only conducted with one target. Future work is necessary to determine whether these observations generalize.

While there are many useful applications for evolving static, single-material 3D objects, this technology is also a stepping stone to evolving objects that can move and that have multiple materials. In future work we will evolve such soft-bodied robots in simulation and transfer them to the physical world. Doing so will enable us to harness the power of evolution and developmental biology to begin to create synthetic creatures that have some of the exciting properties of their natural counterparts.

Acknowledgments

Thanks to Jon Hiller, an NSF Postdoctoral Research Fellowship in Biology to JC (DBI-1003220), NSF CDI Grant ECCS 0941561, and NSF Creative-IT Grant IIS 0757478.

References

Auerbach, J. and Bongard, J. (2010a). Dynamic resolution in the co-evolution of morphology and control. In *Proceedings of Artificial Life XII*.

Auerbach, J. and Bongard, J. (2010b). Evolving CPPNs to grow three-dimensional physical structures. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 627–634. ACM.

Bentley, P. J. (1996). *Generic Evolutionary Design of Solid Objects using a Genetic Algorithm*. PhD thesis, University of Huddersfield.

Bongard, J. and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 829–836.

Clune, J., Ofria, C., and Pennock, R. (2009). The sensitivity of HyperNEAT to different geometric representations of a problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 675–682.

Clune, J., Stanley, K., Pennock, R., and Ofria, C. (2011). On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, To appear.

Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In *Fourth European Conference on Artificial Life*, pages 205–213. The MIT Press.

Hiller, J. and Lipson, H. (2010). Morphological evolution of freeform robots. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 151–152. ACM.

Hornby, G., Lipson, H., and Pollack, J. (2003). Generative representations for the automated design of modular physical robots. *IEEE Transactions on Robotics and Automation*, 19(4):703–719.

Lehman, J. and Stanley, K. (2008). Exploiting open-endedness to solve problems through the search for novelty. *Artificial Life*, 11:329.

Lindenmayer, A. (1968). Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology*, 18(3):280–299.

Lipson, H. (2007). Principles of modularity, regularity, and hierarchy for scalable systems. *Journal of Biological Physics and Chemistry*, 7(4):125.

Lorensen, W. and Cline, H. (1987). Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169.

Miller, J. (2004). Evolving a self-repairing, self-regulating, French flag organism. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 129–139. Springer.

Secretan, J., Beato, N., D'Ambrosio, D., Rodriguez, A., Campbell, A., Folsom-Kovarik, J., and Stanley, K. (2011). Picbreeder: A Case Study in Collaborative Evolutionary Exploration of Design Space. *Evolutionary Computation*, To appear.

Sims, K. (1994). Evolving 3D morphology and behavior by competition. *Artificial Life*, 1(4):353–372.

Stanley, K. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.

Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.

Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.

Wolpert, L. and Tickle, C. (2010). *Principles of Development*. Oxford University Press, 4th edition.