

Novelty Search of Soft Robot Morphologies for Space Exploration

???

ABSTRACT

don't worry about this, we gonna write it at the end

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

General Terms

Theory

Keywords

soft robotics, novelty search, CPPN, HyperNEAT, VoxCAD

1. INTRODUCTION

Motivation, space, small bodies, passive actuation, story of Rosetta/Philae - stupid rigid probe without locomotion, we can do better!

2. BACKGROUND

- gaits at different gravity levels (Ariadna Space Gaits); fixed morphology, rigid body dynamics

2.1 Soft Robots

Soft robotics is a highly promising field of research dedicated to the science and engineering of soft materials in mobile machines. As the name suggests soft robots [26, 16] are made entirely of soft materials mimicking animals or animal-parts that consist only of soft tissue (elephant trunk, tongue, worm, octopus, etc.). Having no rigid parts in their design the degrees of freedom are infinite and the possible ways of motion can become extremely complex. In traditional robotics, joints and rigid parts predefine the space of possible movement and sometimes restrict the robot's locomotion strategy or *gait* to a specific set. In soft robotics, the absence of rigid parts can on the one hand make the

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'15, July 11-15, 2015, Madrid, Spain.

Copyright 2015 ACM TBA ...\$15.00.

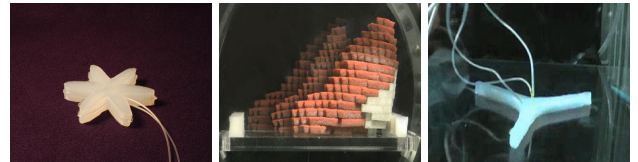


Figure 1: Soft robots can be actuated through air pressure tubes (left), pressure variations (middle), or internal explosions (right).

design of the locomotion strategy exceptionally tortuous, on the other hand the gait alternatives are limitless.

The design and development of soft robotics is not an easy task, while the actuation of such soft structures is the most challenging task. Actuating soft materials can be done in many ways including pneumatic systems [7, 20], hydraulic, internal body explosions, passive actuation triggered by pressure or temperature variations and others [8, 19]. Figure 1 illustrates four different ways that soft robot bodies can be actuated. Regardless traditional ways of actuating soft material robots, three dimensional printing is now giving the freedom for multi-material structures to be created, which also explodes the number of possibilities for the design of a soft structure robot. Autonomously actuated soft robots [25] (see Fig. 2) can also be designed having multiple advantages over rigid body robots such as resistance under extreme temperatures and the capability of locomotion on terrains of variant types.

Although soft robotics research field is in an early stage, it is growing fast. Some of the characteristics that make soft robots interesting to explore are the infinite number of de-



Figure 2: Autonomously actuated soft robot [25], it is able to withstand extreme temperatures and variant terrain types.

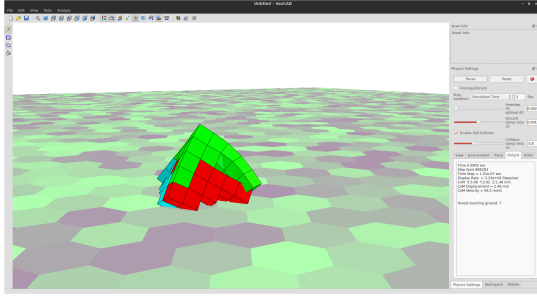


Figure 3: VoxCAD (Voxel CAD), a cross-platform open source voxel modeling and analyzing software.

degrees of freedom and the variety of materials (mostly elastic) that can be used, in the contrary to rigid robotics that are mostly made out of metals and plastic. Nevertheless, structure design and control of soft robotics remain challenging mostly because of their soft bodies can only be represented in continuous state spaces, where only analytic methods can be proven successful.

2.2 Evolution of Soft Robots

Topological optimization techniques can be applied to soft robots [5] for producing functionality in the design. Evolution of soft material robots as it was shown in [3], can result in soft robots able to produce locomotion. The possibility of evolving these soft structures using an indirect encoding was of interest to be exploited by [1]. A powerful generative encoding, CPPNs [22], was used to generate soft voxel-formed three-dimensional structures, coupled with the use of NEAT algorithm which ensures the increasing complexity of the networks produced. The superiority of this kind of generative encoding was verified against direct encoding, showing how CPPNs can take advantage of their geometrical properties. Evaluation was done by a simple displacement measure while evolution tended to evolve different kinds of locomotion strategies and morphologies as the fitness function was penalized for different parameters. Furthermore, it has been shown that evolving morphologies (CPPNs) in lower resolutions and then applying the same networks for higher resolution structures can be beneficial, since the locomotion behaviors in lower structures also apply in higher saving computational time. An earlier work [6], apart from the generative encoding of CPPNs, made use of *Gaussian Mixture* and *Discrete Cosine Transform* to produce amorphous soft body structures. The simultaneous evolution of soft robot morphology and control was also investigated by recent work [17]. Some aspects of soft robot evolution were verified in this work, namely muscle placement and muscle-firing patterns can be evolved given a fixed body shape and fixed material properties. Furthermore, material properties can be co-evolved alongside locomotion strategies. Finally, a developmental encoding was introduced, allowing more complex parts to be added to soft robotic structures during the evolution.

3. METHODOLOGY

3.1 VoxCAD simulator

Most work to simulate interactions and deformations within

and between soft material bodies are mostly focused on the graphical part of the problem [2] sacrificing the accuracy of the simulation [24]. Three dimensional meshes [15] can represent these bodies including the dynamics of their materials.

A recent work, *VoxCad* simulator [4], is focusing mostly on the physics side of the soft material interactions not at the expense of a low frame rate. VoxCad is a modeling and analyzing open-source software that can simulate soft material deformations and interactions. In Figure 3 the graphical user interphase of VoxCad software is presented during the simulation of the soft body robot in the simulator.

VoxCad cannot model and simulate three dimensional meshes, yet a lattice is used to represent the 3D workspace where voxels (three dimensional pixels) can be assigned different materials. Materials themselves are passive and cannot actuate without external trigger. In this simulator this external force that can actuate the materials is the temperature of the environment. The main variables of the environment is the base, the amplitude and the period of the temperature. Materials have properties such as the elasticity of the material, density, Poisson's ratio, coefficient of thermal expansion (which determines how materials will be expanded in respect to the environment's temperature), temporal phase in respect to the temperature period, and the ground friction coefficients. Materials can also be mixed together to create a new type of material. Furthermore, gravity acceleration of the environment can vary.

3.2 Compositional Pattern-Producing Networks

Compositional pattern-producing networks [22] or CPPNs are artificial neural networks with an extended set of activation functions (see Fig. 4). This set of activation functions include repetitive, symmetrical, and linear functions as it is shown in the previous figure. Results by this encoding show that regular patterns can be produced in this generative mapping from the genotype to the phenotype space. CPPNs can generate phenotypes that can be interpreted as distributions of points in a multidimensional Cartesian space. The genotype (CPPN) can then be queried for each coordinate of the space and gives the phenotype representation of the genotype in multiple resolutions. In the same fashion, images can be constructed using CPPNs, where pixel coordinates are queried to the network and the grayscale or RGB values can be taken by the outputs of these networks.

Figure 5 illustrates how the mapping between the genotype and phenotype is done using this type of generative encoding (CPPNs). A major asset of CPPNs is that they

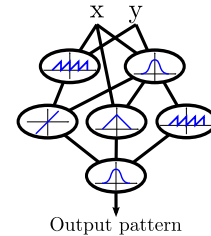


Figure 4: Compositional pattern-producing networks have identical network structure with artificial neural networks while they make use of a canonical set of activation functions.

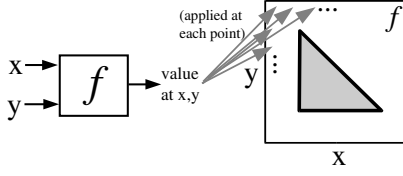


Figure 5: CPPNs work as a function f that is being queried for the whole n -dimensional Cartesian space in which space the phenotype is mapped, in this case the phenotype is the triangle in a two-dimensional space, figure taken by [22].

can generalize in all resolutions. Considering the previous figure (see Fig. 5), the CPPN is queried for all x, y coordinates of the phenotype two dimensional Cartesian space. The step of x, y sampling can be determined by the problem, since the inputs of the CPPN are the normalized coordinates $x, y \in [-1, 1]$. Hence, genotypes using this kind of generative encoding can be mapped in every resolution, making this process straightforward to generalize. As the space of the phenotype becomes larger, a generative encoded solution (CPPN) is not affected by the increasing dimensions of the problem, a constraint that heavily affects direct encoding.

3.3 Neuroevolution of Augmented Topologies

Neuroevolution of augmented topologies (NEAT) as it was first introduced by [21] is a neuroevolution method to evolve artificial neural networks. A major advantage of this method is that alongside the weights it also evolves the topologies of the networks within the population.

Originally, neuroevolution methods were developed to capture difficult sequential decision making, as well as to control problems. The sensory information is the input of these neural networks and decisions are the outputs. NEAT is yet another method for evolving ANNs where a few extra features are added, enables finding solutions in more demanding problems. NEAT starts the evolution process with a population of networks with simple topologies. Through the generations instead of just fixing the weights of the networks' connections, topologies are becoming more complex allowing nodes and links to be added. Meaning that during the evolution, more complicated networks will be produced, this *complexifying* technique leads to capturing more demanding solutions as it offers enough freedom to the evolution.

Several aspects of this method worth mentioning where *speciation* is one of the most important. Speciation is the procedure that protects new *species* until they have enough time to evolve before comparing them with the rest of the population. For two individual genotypes (ANNs) to belong to the same species their network topology must be similar, meaning that a threshold is set and a function determines the numeric value of two network topologies' similarity. Two different genotypes (ANNs) can share shame genes (topologies within the network) and their compatibility (genotype similarity) is given by:

$$\delta = c_1 \frac{E}{N} + c_2 \frac{D}{N} + c_3 \bar{W} \quad (1)$$

where E is the number of *Excess* genes (genes that do not match and they do not occur in parents' genotype), D is

the number of *Disjoint* genes (genes that do not match but they occur in parents' genotype), \bar{W} is the average weight difference between *Matching* genes (Identical) and N is the number of genes in the larger genotype used for normalization.

The age of each species protects them for competing in equal terms with more optimized species, giving them in this way time to evolve further towards the objective function.

3.4 CPPN-NEAT

Compositional pattern-producing networks as described earlier are similar computational methods to ANNs in regards to their structure, so one can make use of the *complexifying* property to capture in this way more complex solutions. NEAT method can evolve CPPNs in the place of ANNs, since it only needs few modifications.

The resulted method that evolves this generative type of genomes (CPPNs) is called CPPN-NEAT [22] and its only difference in respect to the original NEAT algorithm is the way new nodes are added to the network. The original NEAT algorithm evolves ANNs which are using sigmoid functions at every node, so every new node will carry this function. In the contrary, CPPNs use a variety of functions from a canonical set. CPPN-NEAT assigns a random function from this set to every newly added node.

Experiments [22] have shown that this method can indeed evolve CPPNs capturing in this way solutions in problems with geometrical properties (i.e board games, biped walking, etc.). NEAT is holding the properties of natural evolution as every neuroevolution method. Furthermore, NEAT coupled CPPN encoding can be used to determine the connectivity (topology) of artificial neural networks in a method called HyperNEAT [23].

3.5 Novelty search

Novelty search [9, 11, 10, 18] unlike traditional fitness based search is an alternative way of optimization towards an objective function without having knowledge of this objective. In simple words it is looking for a solution to a problem without knowing how close it is to solve it; fact that turns out to have a major impact to the increased performance of this method in several problem domains.

What novelty search seeks for is how interesting a new solution is in respect to all previously found ones. To define "interesting" we need to move our point of interest into behavior space which is a function of each phenotype, similar to the fitness function. Nevertheless, it fully or partially describes the behavior without directly implying the fitness function. As an example someone can think of a behavior could be defined as the recorded trajectory of a robot which tries to maximize its velocity. Rewarding behaviors of the phenotype that are different from the previously found ones drives the evolution to visit new points in the behavior search space.

One significant point here is that the behavior space in some domains can be limitless. However, a valid behavioral metric can be found excluding behaviors that are meaningless or do not comply with the natural limits of the problem. On the other hand, the search space in the genotype level can also be infinite especially in neuroevolution methods like NEAT where ANNs can grow during the evolution. A bounded space of understandable-valid behaviors is then the key idea of novelty search where increasingly complex

behaviors present to the evolution as the complexity of the genotype grows along.

Multi-objective optimization can also make use of a novelty metric alongside fitness, trying to optimize both at the same time [13]. Another method that exploits the diversity of the produced genomes in order to map the phenotype to the fitness has also been proposed [14].

3.5.1 Behaviours

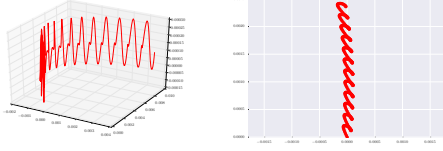


Figure 6: Observed behaviors of the soft robots used for the sparsity computation in novelty search.

Behavior can be defined as the way that a human/machine behaves towards or within an environment. Regarding the evolution of soft robots in the specific simulated environment, a behavior can be defined as the way soft robots behave in respect to their locomotion strategy. Every aspect of the soft robots movement that can be observed can be used to describe their behavior. Previous work [12] in a try to evolve walking three-dimensional virtual creatures used the evolved morphology of the creatures to describe their behavior. Although, in this work comparing the morphology of the evolved soft robots is similar to comparing the chromosome (CPPN) of each individual. Therefore, only the comparison of the observed behavior in phenotype level can lead the evolution towards more complex behaviors.

A straightforward function that determines the goodness of an individual is used in fitness based methods. This measure drives the search in “good” areas of the search space. However, the same measure cannot be used for novelty search. What novelty search looks for is novelty in behavior space. It is expected that behaviors that contain information about the goodness (displacement) of individuals will be more successful than behaviors that include other aspects of the soft robots’ behavior. In cases that behavior does not contain information about the objective, the search for novelty will become random in regards to this objective function.

Behaviors that describe the morphology of the evolved robots have failed [12], since search is then forcing new types of morphologies without caring about the actual target of the evolution, which was the efficient locomotion. To present a similar idea consider a behavior metric that enumerates the number of voxels a soft robot consists of. This is not a well-founded behavior metric, since the search will reward new structures with different number of voxels from previous evolved structures. Therefore, there will not be exploration in the behavior aspect that affects the actual target of the evolution, which is to produce and evolve efficient locomotion strategies.

Figure 6 presents all behaviors used for the novelty metric computation together with the sampling rate of the recorded values during the simulation and their description. For all recorder behavior metrics a constant sampling rate ensures that all signals have the same length. The behaviors designed to describe the strategy and the efficiency of the

evolved locomotion. They contain information that indirectly implies both the objectives of the evolution. *Trajectories* (2D and 3D), incorporate all the needed information such as speed, displacement, and locomotion strategy. To avert from same trajectories in all possible directions trajectories are normalized, meaning that their starting coordinates are always the start of the axes ($< 0, 0, 0 >$) and the point coordinates of the trajectory are rotated so their center of mass is normalized to a specific angle ($\theta = 90^\circ$). To measure the difference of two trajectories the Euclidean distances between coordinates at the same sampling time are measured, so that:

$$\text{First trajectory: } t_i = t_i^1, t_i^2, \dots, t_i^N \quad (2)$$

$$\text{Second trajectory: } t_j = t_j^1, t_j^2, \dots, t_j^N \quad (3)$$

$$\text{Difference: } t_i - t_j = \sum_{n=1}^N \text{dist}(t_i^n, t_j^n) \quad (4)$$

where n is the number of sampled coordinate points and dist is the Euclidean distance. *Pace* is also a very informative behavior metric as it directly measures the speed of the robot. *Voxels touching the ground* can also imply information about the locomotion strategy but not enough about the actual performance regarding the displacement. Hopping robots that move fast can have same behaviors with regards to this metric with hopping robots with zero speed. *Maximum pressure* among the voxels’ connection is yet another behavior metric, pressure is expected to become higher as structures move faster and interactions with the ground eventually getting harder. Finally, *maximum kinetic energy* is a behavior metric that straightly determines the displacement of the voxels in the structure. To compute the difference between two signals, a straightforward method is used. Subtracting the one signal from the other, taking the absolute differences, and summing them up to compute one single value that describes how variant the two signals are. More specifically:

$$\text{First 1-d signal: } s_i = s_i^1, s_i^2, \dots, s_i^N \quad (5)$$

$$\text{Second 1-d signal: } s_j = s_j^1, s_j^2, \dots, s_j^N \quad (6)$$

$$\text{Difference: } s_i - s_j = \sum_{n=1}^N |s_i^n - s_j^n| \quad (7)$$

For all behaviors but trajectories, the Fourier profile of their signals can also be used as an observed behavior. This process of transformation of the one-dimensional signals into frequency space eliminates shifts of signals in time-axis. The discrete Fourier transformation:

$$C_i^k = \sum_{n=1}^N s_i^n e^{-i2\pi kn/N}, \forall k \in \mathbb{Z} \quad (8)$$

For the Fourier transformations of these signals the first twenty coefficients are compared, and the summation of their absolute differences determines the difference of the two behaviors.

$$\text{Difference in frequency: } s_i - s_j = \sum_{k=0}^{R-1} |C_i^k - C_j^k|, R = 20 \quad (9)$$

In this section, different observed behavior metrics have been defined. The similarity or difference of two same type

behaviors can be determined by the equations provided while these measures of difference are used by the sparsity equation (see Eq. ??) to compute the sparseness of a given behavior in the behavior space. Individuals with novel observed behaviors (high sparseness value) are then stored in a list helping the evolution to avoid generating similar behaviors (see Alg. ??).

3.6 Novelty + Fitness-based

3.7 Experimental setup

parameters, gravity

4. RESULTS

novelty (+ fitness) better than fitness examples of cool creatures taxonomy of the evolved creatures at different gravity levels (hoppers, 2-3-4 legs, crawler, tumbleweed)

5. DISCUSSION

what's the use of this? getting inspiration for soft robotic probe / landers (tumbleweed) come back to asteroid scenario (passive motion), would it have saved Philae?

6. CONCLUSIONS

our setup better than results from "unshackling evolution" methodology is suitable to design diverse gaits of soft robots at various gravity levels future work: ensemble of behaviors, very low gravity environments and rotational parameters of small body linked to actuation frequency

7. REFERENCES

- [1] N. Cheney, R. MacCurdy, J. Clune, and H. Lipson. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, pages 167–174. ACM, 2013.
- [2] P. Faloutsos, M. Van De Panne, and D. Terzopoulos. Dynamic free-form deformations for animation synthesis. *Visualization and Computer Graphics, IEEE Transactions on*, 3(3):201–214, 1997.
- [3] J. Hiller and H. Lipson. Automatic design and manufacture of soft robots. *Robotics, IEEE Transactions on*, 28(2):457–466, 2012.
- [4] J. Hiller and H. Lipson. Dynamic simulation of soft heterogeneous objects. *arXiv preprint arXiv:1212.2845*, 2012.
- [5] J. D. Hiller and H. Lipson. Multi material topological optimization of structures and mechanisms. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1521–1528. ACM, 2009.
- [6] J. D. Hiller and H. Lipson. Evolving amorphous robots. In *ALIFE*, pages 717–724, 2010.
- [7] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides. Soft robotics for chemists. *Angewandte Chemie*, 123(8):1930–1935, 2011.
- [8] C. Laschi, M. Cianchetti, B. Mazzolai, L. Margheri, M. Follador, and P. Dario. Soft robot arm inspired by the octopus. *Advanced Robotics*, 26(7):709–727, 2012.
- [9] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.
- [10] J. Lehman and K. O. Stanley. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 103–110. ACM, 2010.
- [11] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.
- [12] J. Lehman and K. O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218. ACM, 2011.
- [13] J.-B. Mouret. Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics*, pages 139–154. Springer, 2011.
- [14] J.-B. Mouret and J. Clune. An algorithm to create phenotype-fitness maps. In *Proc. of the Artificial Life Conf*, pages 593–594, 2012.
- [15] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–54. ACM, 2002.
- [16] R. Pfeifer, M. Lungarella, and F. Iida. The challenges ahead for bio-inspired soft robotics. *Communications of the ACM*, 55(11):76–87, 2012.
- [17] J. Rieffel, D. Knox, S. Smith, and B. Trimmer. Growing and evolving soft robots. *Artificial life*, 20(1):143–162, 2014.
- [18] S. Risi, S. D. Vanderbleek, C. E. Hughes, and K. O. Stanley. How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 153–160. ACM, 2009.
- [19] S. Seok, C. D. Onal, R. Wood, D. Rus, and S. Kim. Peristaltic locomotion with antagonistic actuators in soft robotics. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1228–1233. IEEE, 2010.
- [20] R. F. Shepherd, F. Ilievski, W. Choi, S. A. Morin, A. A. Stokes, A. D. Mazzeo, X. Chen, M. Wang, and G. M. Whitesides. Multigait soft robot. *Proceedings of the National Academy of Sciences*, 108(51):20400–20403, 2011.
- [21] K. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.
- [22] K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.
- [23] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial life*, 15(2):185–212, 2009.
- [24] M. Teschner, B. Heidelberger, M. Müller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Computer*

Graphics International, 2004. Proceedings, pages 312–319. IEEE, 2004.

- [25] M. T. Tolley, R. F. Shepherd, B. Mosadegh, K. C. Galloway, M. Wehner, M. Karpelson, R. J. Wood, and G. M. Whitesides. A resilient, untethered soft robot. *Soft Robotics*.
- [26] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker. Soft robotics: Biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics*, 5(3):99–117, 2008.