Genetic Encoding of Robot Metamorphosis

How to Evolve a Glider with a Genetic Regulatory Network

ANTS 2010, 7th International Conference on Swarm Intelligence

Anne van Rossum (2010)

Introduction

Reconfigurable robotics assigned itself the ambitious task to build not only robot brains, but also *robot bodies*. Artificial organisms can be constructed by connecting smaller robot modules together. Snakes, spiders, chairs, wheels, a prolific number of shapes is possible!

In *Replicator*, a European FP7 project [1], reconfigurable robots are built that maintain their own energy supply. In order to support this, the robots need to be able to morph into proper shapes to overcome an obstacle, switch gears, or connect to a hard to reach power outlet.

Autonomous robotics admits the manifestation of self-driven changes from one shape to the other. Metamorphosis covers the change from snake to spider, wheel to quadruped; a copious space of morphing possibilities.

Engineering *design* of metamorphic body shape transitions might lead to a restricted set of possible changes due to the vastness of the morphing space. It would be appealing to have a methodology in which metamorphosis comes about given a dynamic body shape as the target.

The question arises: *How to evolve self-organized robot metamorphosis?*

Methodology

Morphogenesis, the genesis of a body form, can be encoded by rewriting systems, such as L-systems, evolutionary algorithms including HyperNEAT [2] and *gene regulatory networks* [3].

A gene regulatory network (GRN) is an *indirect encoding* method, and allows for (a) phenotype *compression*, (b) emergent *specialization*, and (c) *coevolution* between body and brain. Most important, a GRN can be used to create dynamic bodies.

Morphodynamics, the emergence of dynamic body shapes, has been researched under the phrase *dynamical blueprint* [4]. This term epitomizes the fact that in this methodology the genome will be continuously read out.

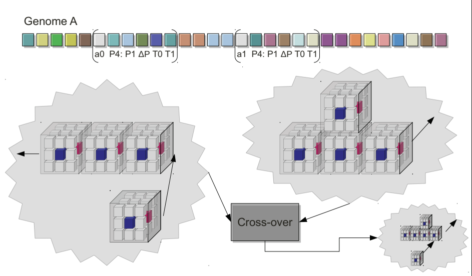

Fig. 2. Recombination between two different dynamic organisms. Left: A glider. Right: A T-shape like organism that loses the right module. Bottom right: After recombination a new organism arises.

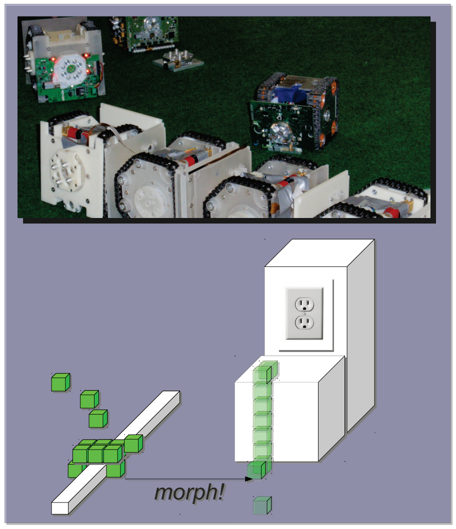

Fig. 1. Illustration of the functional usefulness of metamorphosis. The doggy at the left morphs into a snake at the right to be able to climb on the box and reach the power outlet.

Quick et al. describe the use of a gene regulatory network in a *post-developmental* setting [5]. Khepera robots are evolved such that they exhibit phototactic Braitenberg vehicle behavior. Four protein types are coupled to the sensors and actuators of the robot. A gene regulatory network is evolved that produces proteins in such quantities that the robot demonstrates the phototactic behavior selected for.

Our methodology expands on this work by adding (a) *three-dimensional* spatial encoding, (b) differentiated *multi-cellularity.*

Every robot contains a virtual 3D space in which proteins are diffused from grid cell to grid cell. Proteins also diffuse over robot borders if robots are connected to one another. The position of every robot module in the robot organism defines from which neighbors it receives proteins.

The same GRN operates in every grid cell. However, due to the fact that a cell does have different neighbors, specialization arises.

The *glider*, well-known from the Game of Life, is a prototypical example of a dynamic body shape. A robotic organism reflecting this property is the following: A snake with head, section elements, and tail, grows its head and drops its tail. It does so by adding a module at one side of the organism, and disconnecting a module at the other side. This shape can be seen as a valuable testbench to probe the quality of a genetic algorithm to encode for dynamic body shapes.

http://www.robotcognition.eu
http://replicator.almende.com

Results

The evolutionary experiment uses a population size of 32 swarms. Each swarm consists of 18 robot modules. Every module does have a 3x3x3 cellular grid. The number of protein types is set to 11. The generation count is restricted to 100.

A snake of eight units is evolved using an *incremental scheme*. This scheme uses one robot as seed, inspects its docking requests and builds up on organism from this seed.

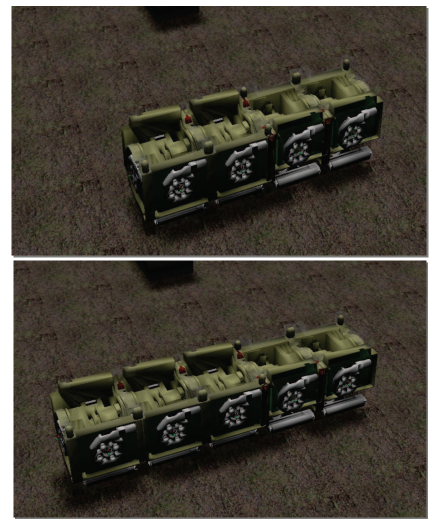

Fig. 3. A "semi-glider" of size 5. Top: Stage before. Bottom: Stage after. This "semi-glider" will drop its newly attached head immediately after it is connected, contrary to a real glider which expands its head and drops its tail. (Robot3D simulator)

A glider consisting out of three units is evolved when bootstrapping the evolutionary search with a genome of an eight unit snake.

Conclusion

Robot metamorphosis, in the form of a potential benchmark case, the glider, has been evolved using a genetic regulatory network (GRN).

A GRN is expressive enough to function as a dynamical blueprint. Further research needs to investigate the metamorphic fitness landscape.

[1] Kernbach, S., Hamann, H., Stradner, J., Thenius, R., Schmickl, T., van Rossum, A.C., Sebag, M., Bredeche, N., Yao, Y., Baele, G., de Peer, Y.V., Timmis, J., Mohktar, M., Tyrrell, A., Eiben, A.E., McKibbin, S.P., Liu, W., Winfield, A.F.T.: *On adaptive self-organization in artificial robot organisms.* In: Proc. of the First IEEE International Conference on Adaptive and Self-adaptive Systems and Applications (IEEE ADAPTIVE 2009). Athens/Glyfada, Greece (2009)

[2] D'Ambrosio, D., Stanley, K.: A novel generative encoding for exploiting neural network sensor and output geometry. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation. p. 981. ACM (2007)

[3] Bongard, J.: *Evolving modular genetic regulatory networks.* In: Proceedings of the 2002 congress on evolutionary computation. pp. 17–21 (2002)

[4] Estévez, N., Lipson, H.: *Dynamical blueprints: exploiting levels of system-environment interaction.* In: Proceedings of the 9th annual conference on Genetic and evolutionary computation. p. 244. ACM (2007)

[5] Quick, T., Nehaniv, C., Dautenhahn, K., Roberts, G.: *Evolving embodied genetic regulatory network-driven control systems.* Lecture notes in computer science pp. 266–277 (2003)