Repeated Structure and Dissociation of Genotypic and Phenotypic Complexity in Artificial Ontogeny

Josh C. Bongard **Rolf Pfeifer**
Artificial Intelligence Laboratory
University of Zürich
CH-8057 Zürich, Switzerland
[bongard|pfeifer]@ifi.unizh.ch

Abstract

In this paper, a minimal model of ontogenetic development, combined with differential gene expression and a genetic algorithm, is used to evolve both the morphology and neural control of agents that perform a block-pushing task in a physically-realistic, virtual environment. We refer to this methodology as artificial ontogeny (AO). It is demonstrated that evolved genetic regulatory networks in AO give rise to hierarchical, repeated phenotypic structures. Moreover, it is shown that the indirect genotype to phenotype mapping results in a dissociation between the information content in the genome, and the complexity of the evolved agent. It is argued that these findings support the claim that artificial ontogeny is a useful design tool for the evolutionary design of virtual agents and real-world robots.

1 Introduction

In the field of evolutionary robotics and artificial life, emphasis is increasingly coming to bear on the question of evolvability: that is, how well the artificial evolutionary system continually discovers agents or robots better adapted to the task at hand (Wagner & Altenberg 1996; Kirschner & Gerhart 1998). It is becoming apparent that modularity, at either the genetic or phenotypic level, or both, is a necessary characteristic of highly evolvable systems (Wagner 1995; Rotaru-Varga 1999; Calabretta *et al* 2000).

Developmental geneticists have made clear that evolved genetic regulatory networks in biological DNA contain master control switch genes, known as *Hox* genes, which orchestrate the transcription of other genes to grow high-level repeated structure, such as the segments in *D. melanogaster* (refer to Gehring & Ruddle (1998) for an overview). It has been shown in a dramatic set of experiments (Lewis 1978) that mutations of *Hox* genes can lead to large-scale but localized changes in phenotype. It has been argued (Raff 1996) that in some cases, differentiation and/or duplication of a feature may allow evolution to co-opt one copy of the feature to perform a different functional role. This process is known as exaptation (Gould & Vrba 1982). A similar mechanism has been shown to have occurred at the gene level (Ohno 1970).

Riedl (1978) demonstrated that the information content of a complex organism is many orders of magnitude higher than that contained in the genome, and has argued that the increased complexity arises from the hierarchical organization of organic units. Raff (1996) has pointed out the same principle holds for the complex processes that take place during ontogeny. Others have argued (Delleart & Beer 1994) that an indirect, developmental genotype to phenotype mapping allows for artificial evolution to discover more complex phenotypes than is possible with direct mappings.

In this paper we introduce an augmented genetic algorithm, in which the genomes are treated as genetic regulatory networks. The changing expression patterns of these networks over time leads to the growth of both the morphology and neural control of a multi-unit, articulated agent, starting from a single unit. We refer to this system as artificial ontogeny (AO), and as is shown in Bongard & Pfeifer (2001), such a system can be used to evolve agents that perform non-trivial behaviours in a physically-realistic, virtual environment, such as directed locomotion in a noisy environment. It is reported here that in agents evolved for a block-pushing task, the morphologies exhibit hierarchical, repeated structure. Evolved agents from previous studies contain repeated structure, however these studies relied on more direct, parametric encoding schemes (Sims 1994; Ventrella 1996; Komosinski & Ulatowski 1999; Lipson & Pollack 2000). Conversely, in studies conducted using developmental encoding schemes, the agents are relatively simple, and do not exhibit any higher-order, repeated structure (Delleart & Beer 1994; Jakobi 1995).

In the next section the morphologies of the evolved agents are explained, the differential gene expression model used to grow them, as well as the method by which neural networks are grown along with the developing morphology of the agent. The following section reports the results of a set of evolutionary runs in which agents are evolved for a block-pushing task, and provides some analysis of the resulting phenotypes and gene expression patterns. The penultimate section discusses the adaptive potential of the AO system, and promising areas of future research. The final section provides some concluding remarks.

2 The Model

In this system, there is a translation from a linear genotype into a three-dimensional agent complete with sensors, actuatable limbs and internal neural architecture, such as in Sims (1994), Ventrella (1996), Komosinski & Ulatowski (1999), Bongard & Paul (2000), and Lipson & Pollack (2000). However unlike these other methods, the genotype to phenotype translation described here takes place via ontogenetic processes, in which differential gene expression, coupled with the diffusion of gene products, transforms a single structural unit in a continuous manner into an articulated agent, composed of several units, some or all of which contain sensors, actuators and internal neural structure.

2.1 Agent Morphology

Each agent evaluated in the physically-realistic simulation is composed of one or more units. For the experiments reported here, spheres are used to represent these units. By scaling up the number of units used to construct an agent, increasingly arbitrary morphologies can be evolved. Each agent begins its ontogenetic development as a single unit. Depending on the changing concentrations of the gene products within this unit, the unit may grow in size, until the radius grows to twice that of the unit's original radius. At this point the unit splits into two units; the radii of both the parent and child units are then reset to the default radius.[1]

Each unit contains: zero to six joints attaching it other units via rigid connectors; a copy of the genome directing development of the given agent; and six diffusion sites. Each of the six diffusion sites are located midway along the six line segments originating at the centre of the sphere, terminating at the surface, and pointing north, south, west, east, up and down. Each diffusion site contains zero or more diffusing gene products and zero or more sensor, motor and internal neurons. The neurons at a diffusion site may be connected to other neurons at the same diffusion site, another diffusion site within the same unit, or to neurons in other units. Each of the components of a unit are described in more detail in the following sub-sections.

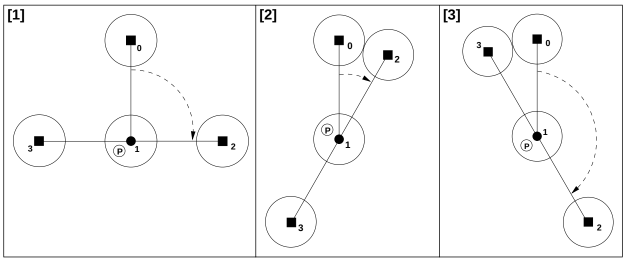

Figure 1: **Architecture of articulated joints** Panels [1] through [3] depict part of an agent's morphology. In this hypothetical scenario, unit 1 split from unit 0, and units 2 and 3 split from unit 1. The black squares represent fused joints; the black circles represent rotational joints. The fused joints connecting units 2 and 3 to unit 1 are not shown for clarity. Rotation occurs through the plane described by the angle between units 0, 1 and 2. Panel [1] shows the configuration of the agent immediately after growth, before activation of the neural network. Unit 1 contains a proprioceptive sensor neuron, which emits a zero signal. In panel [2], unit 1 has rotated counterclockwise, either due to internal actuation or external forces. The proprioceptive sensor in unit 1 emits a nearly maximal negative value. In panel [3], the hinge in unit 1 reaches has rotated clockwise: the proprioceptive sensor now emits a nearly maximal positive signal. Note that the architecture of the agent's morphology precludes the hinge from reaching its rotational limits, and the proprioceptive sensor from generating either a maximally negative or positive signal.

A newly-created unit is attached to its parent unit in one of six possible directions using a rigid connector that maintains a constant distance between the units, even though one or both of the attached units may continue to grow in size. The new unit is placed opposite to the diffusion site in the parent unit with the maximum concentration of growth-enhancing gene product. After a unit splits from its parent unit, the two units are attached with a rigid connector, the ends of which are located in the centres of the two units. The parent unit is fixed to the rigid connector. The new unit is attached to the rigid connector by a one degree of freedom rotational joint. The fulcrum of the joint is placed in the centre of the new unit. Joints can rotate between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ radians of their starting orientation. The axis about which a unit's joint rotates is set perpendicular to the plane described by the parent unit, the child unit, and the first unit to split from the child unit. If no units split from a unit, that unit's rotational joint is removed, and the unit is fixed to the rigid connector it shares with its parent unit. This precludes the evolution of wheels, in which units rotate about their own centre of mass. Fig. 1 illustrates the creation and actuation of an agent's joints in more detail.

[1] Although the agent grows through repeated division of units, and each unit retains a copy of the genome that directs the agent's growth, the units used in this model are not to be equated with the biological concept of a cell, such as in the AES system (Eggenberger 1997), nor are they equivalent to the units employed in the parametric models mentioned above. Rather, repeated division is a useful abstraction that allows for a relatively continuous transition from a single unit into a fully developed agent composed of many such units.

The agent's behaviour is dependent on the real-time propagation of sensory information through its neural network to motor neurons, which actuate the agent's joints.

There are three types of sensors that artificial evolution may embed within the units of the agent: touch sensors, proprioceptive sensors, and light sensors. Touch sensor neurons return a maximal positive signal if the unit in which they are embedded is in contact with either the target object or the ground, or a maximal negative signal otherwise. Proprioceptive sensors return a signal commensurate with the angle described by the two rigid connectors forming the rotational joint within that unit (refer to Fig. 1). Light sensor neurons return a signal that is linearly correlated to the distance between the unit in which the sensor is embedded and the target object in the environment. The light sensors are not physically simulated, but calculated geometrically.

The agent achieves motion by actuating its joints. This is accomplished by averaging the activations of all the motor neurons within each unit, and scaling the value between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. Torque is then applied to the rotational joints such that the angle between the two rigid connectors forming the joint matches this value. The desired angle may not be achieved if: there is an external obstruction; the units attached to the rigid connectors experience opposing internal or external forces; or the values emitted by the motor neurons change over time. Note that failure to achieve the desired angle may be exploited by evolution, and may be a necessary dynamic of the agent's actions. If a unit contains no motor neurons, the rotational joint in that unit is passive.

Internal neurons can also be incorporated by evolution into an agent's neural network, in order to propagate signals from sensor to motor neurons. Two additional neuron types are available to evolution. Bias neurons emit a constant, maximum positive value. Oscillatory neurons emit a sinusoidal output signal. The summed input to an oscillatory neuron modulates the frequency of the output signal, with large input signals producing an output signal with a high frequency, and low input signals producing a low frequency output signal.

2.2 Differential Gene Expression

Unlike the recursive parametric encoding schemes mentioned above, each genome in the AO system is treated as a genetic regulatory network (Kauffman 1993, Jakobi 1995, Eggenberger 1997 and Reil 1999), in which genes produce gene products that either have a direct phenotypic effect or regulate the expression of other genes.

For each genome to be evaluated in the population, it is first copied into the single unit from which the eventual fully-formed agent develops. The genome is then scanned by a parser, which marks the site of promotor sites. Promotor sites indicate the starting position of a gene along the genome. A value in the genome is treated as a promotor site if the value is below $\frac{n}{l}$, where n is the average number of genes that should appear within each initial random genome, and l is the length of genomes in the initial, random genetic algorithm population. This is done so that, given a starting population of random genomes, each genome will contain, on average, the desired number of genes. In the results reported in the next section, $l = 100$ and $n = 10$, causing values between 0.00 and 0.10 to serve as promotor site indicators.

Fig. 2 provides a pictorial representation of a genome directing the growth of an agent. The seven floating-point values following a gene's promotor site supply the parameter values for the gene. If the first value ($P1$ in Fig. 2) is less than 0.5, gene expression is repressed by presence of the gene product which regulates its expression; otherwise gene expression is enhanced by presence of its regulating gene product. The second value ($P2$ in Fig. 2) indicates which of the 24 possible gene products regulates the gene's expression. The third value ($P3$ in Fig. 2) indicates which of the 24 possible gene products is produced if this gene is expressed. The fourth value ($P4$ in Fig. 2) indicates which of the 6 gene product diffusion sites the gene product is diffused from if this gene is expressed. The fifth value ($P5$ in Fig. 2) indicates the concentration of the gene product that should be injected into the diffusion site if the gene is expressed. The sixth and seventh values ($P6$ and $P7$ in Fig. 2) denote the concentration range of the regulating gene product to which the gene responds. If the concentration of the regulating gene product to which the gene responds is within this range, and the gene is enhanced by presence of its regulating gene product, the gene is expressed; otherwise, gene expression is repressed. Genes that are repressed by their regulating gene product are expressed if the gene product's concentration is outside the denoted range, and repressed otherwise.

After the genes in the genome have been located, the originating unit of the agent to be grown is injected with a small amount of gene product at diffusion site 1. Due to gene product diffusion, a gradient is rapidly established in this first unit, among the 6 diffusion sites. This is analogous to the establishment of a gradient of maternal gene product in fruit flies, which leads to the determination of the primary body axis (Anderson 1984), and breaking of symmetry in early embryogenesis. It can be seen from Fig. 3 that the degree of symmetry in evolved agents varies, and is under evolutionary control.

As the injected gene product diffuses throughout the unit, it may enhance or repress the expression of genes along the genome, which in turn may diffuse other gene products. There are 24 different types of gene products. Two affect the growth of the unit in which they diffuse. At each time

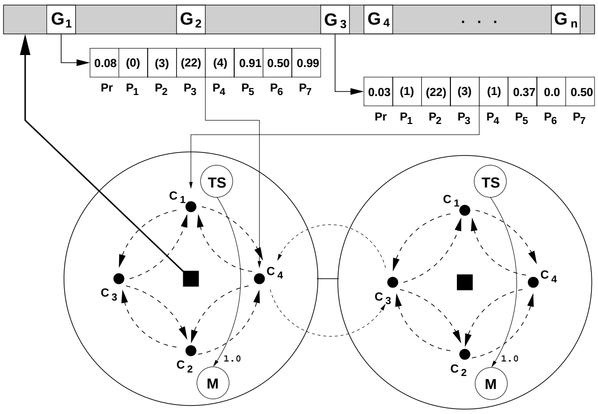

Figure 2: **Ontogenetic interactions in a developing agent** Two structural units of an agent are shown above, but only displayed in two dimensions for clarity. For this reason, only four of the six gene product diffusion sites are shown; the other two lie at the top and bottom of the spherical units. The genome of the agent is displayed, along with parameter values for two genes. The values in parentheses indicate that these values are rounded to integer values. Gene G_1 indicates that it is repressed (parameter P_1) by concentrations of gene product 3 (P_2) between 0.5 and 0.99 (P_6, P_7). Otherwise, it diffuses gene product 22 (P_3) from gene product diffusion location 4 (P_4), indicated in the diagram by C_4. Note that genes G_1 and G_3 emit gene products which regulate the other's expression. The thick dotted lines indicate gene product diffusion between diffusion sites within a unit; the thin dotted lines indicate gene product diffusion between units. Both units contain a touch sensor neuron (TS) and a motor neuron (M) connected by excitatory synapses.

step of the development phase, the difference between the concentration of these two chemicals is computed. If the difference is positive, the radius of the unit is increased a small increment; if the difference is negative, the unit does not grow in size. Thus these two chemicals function as growth enhancer and growth repressor, respectively. If the radius of a unit reaches twice that of its original radius, a split event is initiated. The radius of the parent unit is halved, the gene product diffusion site with the maximum concentration of growth enhancer is located, and a new unit is attached to the parent unit at this position. Half of the amounts of all gene products at this diffusion site are moved to the neighbouring diffusion site in the new unit. A copy of the genome is assigned to the new unit. The gene expression patterns of the parent and child units are now independent, except for indirect influence through inter-unit diffusion of gene products.

There are then 17 other chemicals which affect the growth of the agent's neural network, and are explained in the next section. Finally, five gene products have no direct phenotypic effect, but rather may only affect the expression of other genes. That is, concentrations of these gene products at diffusion sites can enhance or repress gene expression in that unit (like the other 19 gene products), but cannot modify neural structure, or stimulate or repress the growth of that unit.

All 24 gene products share the same fixed, constant diffusion coefficients. For each time step that a gene emits gene product, the concentration of that gene product, at the diffusion site encoded in the gene, is increased by the amount encoded in the gene (which ranges between 0.0 and 1.0), divided by 100. All gene product concentrations, at all diffusion sites, decay by 0.005 at each time step. Gene products diffuse between neighbouring diffusion sites within a unit at one-half this rate. Gene products diffuse between neighbouring units at one-eighth the rate of intra-unit diffusion.

2.3 Neural Growth

Cellular encoding (Gruau 1996) has been incorporated into our model to achieve the correlated growth of morphology and neural structure in a developing agent. Cellular encoding is a developmental method for evolving both the architecture and synaptic weights of a neural network. The process involves starting with a simple neural network of only one or a few neurons, and iteratively or recursively applying rewrite rules that modify the architecture or synaptic weights of the growing network.

In our model, for each new unit that is created, including the first unit, a small neural network is created as follows: A touch sensor neuron (TS) is placed at diffusion site 1, a motor neuron (M) is placed at diffusion site 2, and a synapse with a weight of 1.0 is connected from the sensor neuron to the motor neuron (refer to Fig. 2). When a unit undergoes a split event, any neurons at the diffusion site where the split event was initiated are moved to the neighbouring diffusion site of the new unit. For example, if a unit splits, and the new unit is attached near its northern face, all the neurons in the northern diffusion site of the parent unit are moved to the southern diffusion site in the new unit. Neurons may also move from one diffusion site to another within a unit, depending on the concentrations of gene products at those sites. The combination of these dynamics may lead to the directed migration of neurons across the units as they divide. As they migrate, synapses connecting these neurons are maintained: although this process is different from the neural growth cone model (in which biological neurons innervate distant cells using exploratory synaptic outgrowths (Kater 1990)) and instantiations of this model (Delleart & Beer 1994; Jakobi 1995), it does allow for neurons in distant units to remain connected.

Each of the 17 gene products responsible for neural development correspond to one rewrite operation that modifies local neural structure. At each diffusion site, two pointers are maintained: the first pointer indicates which synapse will undergo any synaptic modification operations; the second pointer indicates which neuron will undergo any neuronal modification operations. The 17 rewrite rules correspond to serial and parallel duplication of neurons; deletion of neurons and synapses; increase and decrease of synaptic weight; duplication of synapses; neuron migration within a unit; changing of the afferent and efferent target of synapses; and changing of neuron type. If the concentration of one of these 17 gene products at a diffusion site exceeds 0.8, and there is neural structure at that site, the corresponding operation is applied to the neural structure there. Once development is complete, the neural network that has grown within the agent is activated. At each time step of the evaluation period, the input to each neuron is summed, and thresholded using the activation function $\frac{2}{1+e^{-s}} - 1$, where x is the neuron's summed input. Neuron values can range between 1 and -1. Using this neural development scheme, the AO system is able to evolve dynamic, recurrent neural networks that propagate neural signals from sensor neurons to motor neurons distributed throughout an agent's body.

3 Results and Analysis

The evolutionary runs reported in this section were conducted using a variable length genetic algorithm; the genomes were strings of floating-point values ranging between 0.00 and 1.00, rounded to a precision of two decimal places. A population size of 200 was used, and each run lasted for 200 generations. All genomes in the initial random population have a starting length of 100 values. The mutation rate was set to produce, on average, random replacement of a single value for each new genome. Unequal crossover was employed, which allowed for gene duplication and deletion. Tournament selection, with a tournament size of 2, was used to select genomes to participate in crossover.

As in Bongard & Pfeifer (2001), agents are evaluated in a physically-realistic virtual environment using a commercially available physics-based simulation package[2]. Each genome in the population is evaluated as follows: The genome is copied into a single unit, which is then placed in a virtual, three-dimensional environment. A target cube is placed 20 units[3] to the north of the unit; the sides of the cube are 70 units long. Morphological and neural development is allowed to proceed, as described in the previous section, for 500 time steps. After the development phase, the neural network is activated, and the agent is allowed to operate in its virtual environment for 1000 time steps. The fitness of an agent is given as $\sum_{i=2}^{1000} n(t(i-1)) - n(t(i))$, where $n(t(i))$ is the northern distance of the centre of the cube from the origin at time t. Thus the agent is rewarded for reaching the cube as fast as possible, and pushing it as far as possible. By making the cube much larger than the units comprising an agent, we can exert indirect selection towards large agents: agents must have a large mass in order to exert a large force against the cube. Agents a) and b) in Fig. 3 depict the morphologies of the most fit agents from two independent runs. Agents c) and d) were the most fit agents at generation 110 and 130 of the run shown in Fig. 4.

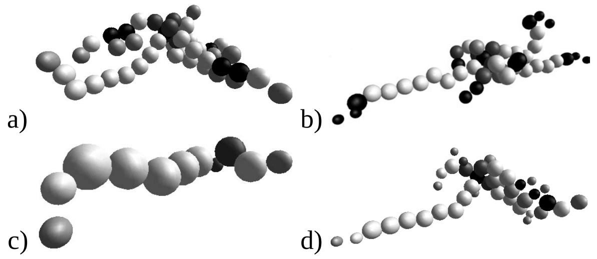

Figure 3: **Four agent morphologies** The block is not shown in the figure for the sake of clarity, but lies just to the left of the agents. The rigid connectors are also not shown. The white units indicate the presence of both sensor and motor neurons within that unit. The light gray units indicate the presence of both sensor and motor neurons in that unit, but the one or more motor neurons do not actuate the rotational joint in that unit either because there are no input connections to the motor neuron, or because there is no joint within this unit. The dark gray units indicate the presence of sensor neurons, but no motor neurons. The black units indicate the unit contains neither sensor nor motor neurons.

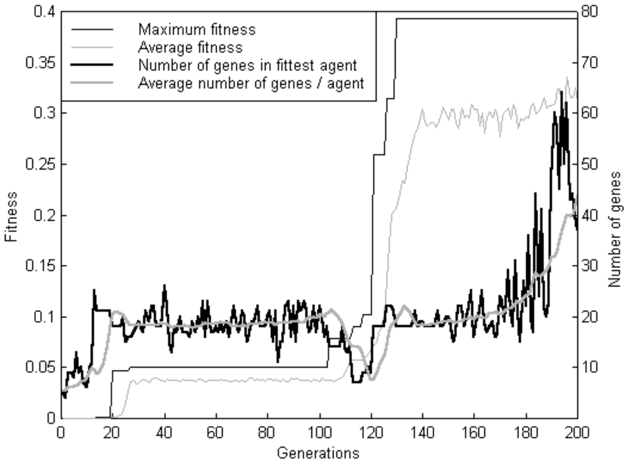

Figure 4: **Results from a typical run.** Genome length was found to be roughly proportional to the number of genes, and is not plotted.

In order to detect the presence of hierarchical, repeated structure in evolved agents, the local neural structure within units was used as a signature to distinguish between units. For instance in agent a) in Fig. 3, the two neighbouring units that have lost their motor and sensory capabilities are repeated twice. In the right-hand agent, the three most distal units in the three main appendages have also lost their motor and sensory capabilities.

The most fit agent from each of the nine evolutionary runs

[2] MathEngine PLC, Oxford, UK, `www.mathengine.com`

[3] Spatial distance in the physics-based simulator is relative; we treat a 'unit' as equal to the default radius of a newly-created unit.

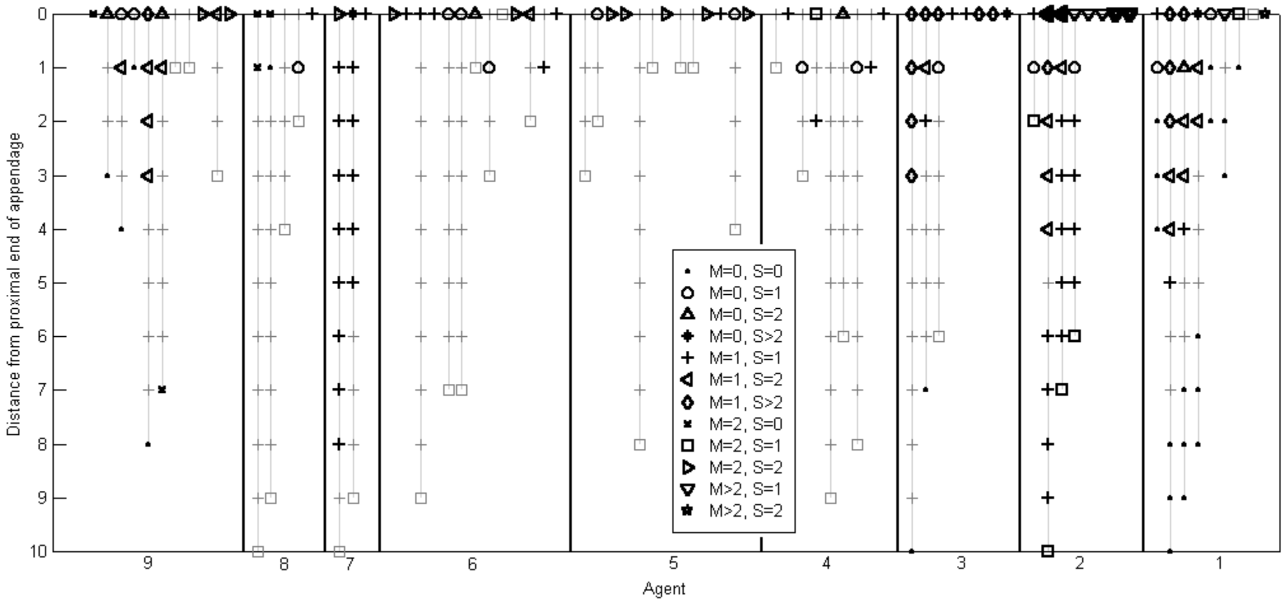

Figure 5: **Neural composition of nine evolved agents** Each symbol indicates the number of motor and sensor neurons in a structural unit. Neural structure is only reported for units that are part of an appendage. Units comprising an appendage are linked by gray lines. Gray symbols indicate no rewrite rules have been applied to the neural structure in that unit; black symbols indicate units in which genetic manipulation of local structure has occurred. The gene expression patterns of the four units indicated by bold symbols is shown in Fig. 6. Agent 1 corresponds to agent b) in Fig. 3.

was extracted, and the number of motor and sensor neurons in each unit of each appendage was counted. The tallies for each unit are reported in Fig. 5. Because the units comprising an agent are organized as directed trees, appendages can be determined as follows: for each terminal unit in the agent, traverse up the tree until a unit is found with more than one child unit. The units that were traversed, minus the last one counted, comprise an appendage.

Finally, the gene expression patterns of four units are reported in Fig. 6. Units a) and c) give rise to appendages with similar patterns of local neural structure, and themselves have similar internal neural structure. Units b) and d) do not give rise to further structure, and have similar neural structure. This structure is different from units a) and c). The four units are indicated in bold in Fig. 5. Units a) through d) all split from the same parent unit during ontogeny, but appear at increasingly later times during the agent's development.

4 Discussion and Future Work

Fig. 4 indicates that no agent is able to push the block until generation 20; this event is accompanied by a doubling in the number of genes carried by these more fit agents. However, the gene complement of agents does not increase considerably during the rapid fitness increase which occurs around generation 120. Agents c) and d) in Fig. 3 indicate that this fitness increase was accomplished by a radical increase and reorganization of the agent's morphology and neural control. This suggests that the AO system is exhibiting that predicted property of indirect encoding schemes, that is, large increases in phenotypic complexity[4] without corresponding large increases in genome size.

Fig. 5 indicates that invariably, evolution converges on agents that exhibit hierarchical repeated structure. This can be seen most clearly in the first agent in Fig. 5, in which the first agent contains three similar appendages with three distal units each containing neither motor nor sensor neurons. Moreover, Fig. 5 indicates that genetic changes to local neural structure can be repeated both within an appendage—as seen by the deletion of function in the three distal units—and across appendages—as seen by the triple deletion of function repeated in three different appendages.[5] In other words, agents tend to have appendages in which local neural structure is repeated along the length of the appendage, and appendages themselves are repeated. It is important to note that this structure—which we, as observers, consider hierarchical, repeated structure—is the result of the complex, dynamical interplay between the evolved genetic regulatory networks, the developmental process, and the selection pressure exerted on the evolving

[4] In this context, complexity is simply taken as the number and organization of units, and variation in local neural structure within those units.

[5] From visual inspection of these agent's behaviours, it seems as if these appendages use a whiplike motion, requiring strong actuation at the proximal end and little or no actuation at the distal end.

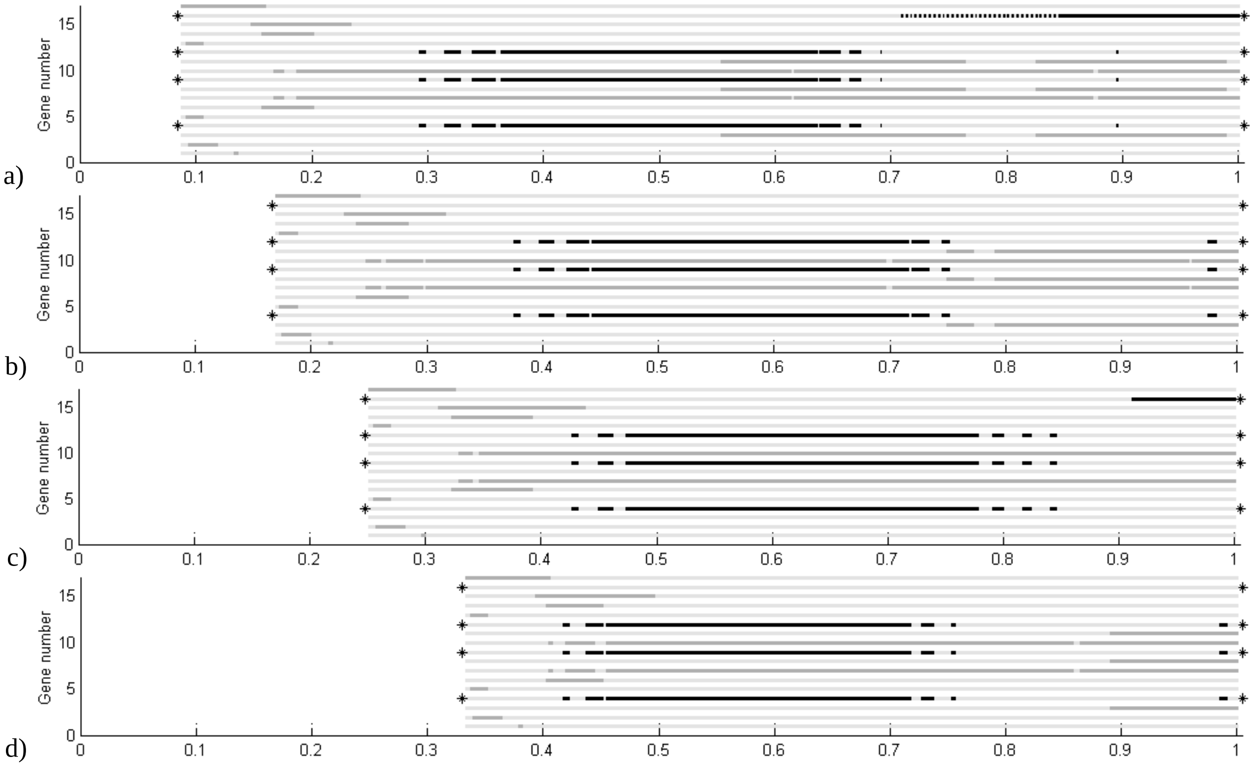

Figure 6: **Gene expression patterns for four units.** Dark gray and light gray bands correspond to periods of gene activity and inactivity, respectively. Four genes are marked by asterisks; the expression pattern of these genes is similar in units a) and c), but different in units b) and d). The expression times of these genes are darkened for clarity. Genes that are always on or always off during ontogeny are not shown. Note the evolved gene families, which have similar expression patterns.

population. This suggests that the study of genetic regulatory networks should not be conducted in isolation, but rather in the context of embodied agents evolved for a specific task. This would then give us a clearer picture of how both natural and artificial evolution shape such regulatory networks over time.

Finally, Fig. 6 indicates that the units that give rise to similar appendages have similar gene expression patterns, even though they appear at different times during ontogeny. Similarly, the gene expression patterns of two other units, which appear at roughly the same time as the other two units, correspond. However, the gene expression patterns are different between these two pairs of units. This is shown by the expression of the first marked gene in units a) and c), but not in b) and d); a short expression band for the other three marked genes appears during late ontogeny in units b) and d), but not in a) and c). This indicates that, even though all four of these units originated from the same parent unit, and at roughly the same time during ontogeny, the units which gave rise to appendages have a shared pattern of expression that differs from the pair that does not give rise to appendages. This result suggests that future studies might uncover one or a small set of genes that lead to the growth of higher-order structure when active, but repress such growth when inactive. These genes would serve as analogues of *Hox* genes in biological organisms, and would indicate that such genes are the natural result of evolution when coupled with ontogeny and differential gene expression. Our future studies will also include more detailed analysis of the evolved genetic networks.

5 Conclusions

To conclude, this paper has demonstrated that a minimal model of biological development, coupled with a genetic algorithm that allows for gene duplication and deletion, is sufficient to evolve agents that perform a non-trivial task in a physics-based virtual environment. Moreover, this system—referred to as artificial ontogeny—is sufficient to produce hierarchical, repeated phenotypic structure. In addition, it has been shown that the inclusion of differential gene expression in artificial ontogeny dissociates the information content of the genome from the complexity of the evolved phenotype.

Both of these properties point to the high evolvability of the AO system: both the production of hierarchical, repeated organization and the dissociation of genotypic and phenotypic complexity are necessary if artificial evolution is to prove useful for the design of robots that solve increasingly complex tasks, the ultimate goal of evolutionary robotics research.

References

J. Bongard and C. Paul (2000). Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution. In *Proceedings of the Sixth International Conference on Simulation of Adaptive Behaviour*, pp. 420–429. MIT Press.

J. Bongard and R. Pfeifer (2001). Evolving complete agents using artificial ontogeny. To appear in *Proceedings of The First International Workshop on Morpho-functional Machines*, Springer-Verlag, Berlin.

K. V. Anderson and C. Nüsslein-Volhard (1984). Information for the dorso-ventral pattern of the *Drosophila* embryo is stored in maternal mRNA. In *Nature* **311**:223–227.

R. Calabretta, S. Nolfi, D. Parisi and G. P. Wagner (2000). Duplication of modules facilitates the evolution of functional specialization. In *Artificial Life* **6**(1):69–84. Cambridge, Mass: MIT Press.

F. Delleart, and R. D. Beer (1994). Toward an evolvable model of development for autonomous agent synthesis. In *Artificial Life IV*, 246–257. MIT Press.

P. Eggenberger (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In *Proceedings of the Fourth European Conference on Artificial Life*, 205–213. Berlin: Springer-Verlag.

W. J. Gehring and F. Ruddle (1998). *Master Control Genes in Development and Evolution: The Homeobox Story (Terry Lectures)*, New Haven: Yale University Press.

S. J. Gould and E. S. Vrba (1982). Exaptation—a missing term in the science of form. In *Paleobiology* **8**:4–15.

F. Gruau, D. Whitley, and L. Pyeatt (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. In Proceedings of the First Genetic Programming Conference, 81–89. MIT Press.

N. Jakobi (1995). Harnessing morphogenesis. Presented at *The International Conference on Information Processing in Cells and Tissues*, Liverpool, UK.

S. B. Kater and P. B. Guthrie (1990). Neuronal growth cone as an integrator of complex environmental information. In *Cold Spring Harbor Symposia on Quantitative Biology, Volume LV*, 359–370. Cold Spring Harbor Laboratory Press.

S. A. Kauffman (1993). *The Origins of Order*, Oxford, UK: Oxford University Press.

M. Kirschner and J. Gerhart (1998). Evolvability. In *Proc. Nat. Acad. Sci* **95**:8420-8427.

M. Komosinski, and S. Ulatowski (1999). Framsticks: Towards a simulation of a nature-like world, creatures and evolution. In: *Proceedings of 5th European Conference on Articial Life*, 261–265. Springer-Verlag.

E. B. Lewis (1978). A gene complex controlling segmentation in Drosophila. In *Nature* **276**:565-570.

H. Lipson and J. B. Pollack (2000). Automatic design and manufacture of artificial lifeforms. In *Nature* **406**:974–978.

S. Ohno (1970). *Evolution by Gene Duplication*. New York: Springer Verlag.

R. A. Raff (1996). *The Shape of Life*. Chicago: The University of Chicago Press.

T. Reil (1999). Dynamics of gene expression in an artificial genome–implications for biological and artificial ontogeny. In *Proceedings of the Fifth European Conference on Artificial Life*, 457–466. Springer-Verlag.

R. Riedl (1978). *Order in Living Organisms: A Systems Analysis of Evolution*. Chichester: John Wiley & Sons.

A. Rotaru-Varga (1999). Modularity in evolved artificial neural networks. In *Proceedings of the Fifth European Conference on Artificial Life*, 256–260. Spring-Verlag.

K. Sims (1994). Evolving 3D morphology and behaviour by competition. In *Artificial Life IV*, 28–39. MIT Press.

D. Terzopoulos, T. Rabie and R. Grzeszczuk (1996). Perception and learning in artificial animals. In *Artificial Life V*, 313–320. MIT Press.

J. Ventrella, (1994). Explorations of morphology and locomotion behaviour in animated characters. In *Artificial Life IV*, pp. 436–441. MIT Press.

G. P. Wagner (1995). Adaptation and the modular design of organisms. In: *Advances in Artificial Life*, 317–328. Springer Verlag.

G. Wagner and L. Altenberg (1996). Perspective: Complex adaptations and the evolution of evolvability. In *Evolution* **50**(3):967–976.