# Evolving Complete Agents using Artificial Ontogeny

Josh C. Bongard Rolf Pfeifer
Artificial Intelligence Laboratory
University of Zurich
CH-8057 Zurich, Switzerland
e-mail: [bongard,pfeifer]@ifi.unizh.ch

## Abstract

In this report we introduce an artificial evolutionary system, Artificial Ontogeny (AO), that uses a developmental encoding scheme to translate a given genotype into a complete agent, which then acts in a physically-realistic virtual environment. Evolution is accomplished using a genetic algorithm, in which the genotypes are treated as genetic regulatory networks. The dynamics of the regulatory network direct the growth of the agent, and lead to the construction of both the morphology and neural control of the agent. We demonstrate that such a model can be used to evolve agents to perform non-trivial tasks, such as directed locomotion and block pushing in a noisy environment. It is shown that mutations expressed earlier in development tend to have a more variable morphological and behavioural effect than mutations expressed later in development, which tend to have a less pronounced effect. These results support the hypothesis that ontogeny provides artificial evolution with beneficial mutations that have varying degrees of phenotypic effect, depending on their onset of expression during development. In addition, we evolve agents using a fitness function which indirectly selects for increased size. In these agents we find evidence of functional specialization and repeated, differentiated structure. In the final section we argue that such a system would be a useful tool for the evolutionary design of morpho-functional machines.

## 1 Introduction

There are three major obstacles challenging the investigation of the interdependence between an adaptive agent's morphology, neural control and environment in the field of artificial intelligence. First, the internal processing bias in classical artificial intelligence has tended to place an inordinate emphasis on neural processing, to the detriment of the other aspects of an agent. Second, technological limitations have largely prohibited the investigation of the behavioural effect of changing the morphology of real-world robots. Finally, the computational resources required to evolve both the morphology and neural control of simulated agents, in a physically-realistic environment, have, until recently, been beyond the reach of most artificial intelligence research laboratories.

However, the emerging field of embodied cognitive science [18, 3] is beginning to challenge the computational biases of classical artificial intelligence. The field of morpho-functional machines is producing examples of robots with flexible morphological configurations [15, 12, 8]. Finally, several examples of evolved complete agents in physically-

realistic environments have recently appeared in the literature [23, 14, 2, 16], either extending or complementing the work conducted by Karl Sims in 1994 [21]. This paper contributes to the third stream of investigation by demonstrating that the close interdependence between morphology and behaviour allows Artificial Ontogeny—that is, an artificial evolutionary system with differential gene expression during growth—to evolve complete agents that solve a non-trivial task.

All of the evolved agents reported in [21, 23, 14, 2, 16] were evaluated within a physically realistic virtual environment. The agents reported in [16] were then constructed as real-world robots. Terzopoulos [22] and Ijspeert [9] evolved neural controllers for agents which acted within a physically-realistic environment. By using a physically-realistic environment, artificial evolution is free to tune both the morphology and neural control of the evolving agents such that they exploit the physics of their environment. For example, we reported in [2] that only a fraction of the joints in some agents evolved for locomotion are actuated. The other joints are passive, and the forward momentum of the agent—along with that agent's particular morphology—moves these joints in an appropriate manner during locomotion. This demonstrated that artificial evolution can make use of passive dynamics [17]. As in [2], the experiments reported in this paper were conducted using a commercially available physics-based simulation package[1].

Eggenberger [5] used an evolutionary method for evolving agents that grow via differential gene expression, but the agents lacked neural structure, and were only evaluated as to their body shape. This report extends the work of Eggenberger by using genetic regulatory networks to evolve both the morphology and neural control of complete agents. The aim of this work is not to produce a detailed model of biological development, but rather to abstract just those developmental mechanisms from nature that makes the AO system highly evolvable [24, 13]. That is, given an arbitrary task, the abstracted biological mechanisms lend the AO system a relatively high probability of finding an agent in the search space that performs the task well.

Section 2 provides details about the genetic regulatory networks, as well as other aspects of the AO system. Section 3 demonstrates some of the agents that were evolved, as well as results detailing how selection pressure tends to modify differential gene expression patterns to produce the evolved agents. Section 4 provides an analysis of the results, and the final section provides some concluding remarks and promising avenues for future investigation.

# 2 The Model

In this section we introduce an artificial evolutionary method which we term Artificial Ontogeny (AO). In this method, there is a translation from a linear genotype into a three-dimensional agent complete with a set of sensors, actuated and/or passive limbs, and internal neural architecture, such as in [21, 23, 2, 16]. However unlike these other methods, the genotype to phenotype translation 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 structural units, some or all of which contain sensors, actuators and internal neural

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

structure. If we wish to evolve both the morphologies and controllers of robots, biological ontogeny seems a useful (and, at the moment, the only) guide upon which evolution can coordinate the construction of the various morphological and neural structures necessary for a functioning agent. First we describe the mechanical, morphological aspects of the agents that are evolved. This is followed by a description of the differential gene expression model, and how this affects the ontogenetic development of an agent. The final sub-section details how neural structure grows in time with the growing body of the agent.

## 2.1 Agent Morphology

Each agent evaluated in the physically-realistic simulation is composed of one or more structural units. For the experiments reported here, spheres are used to represent these units. Spheres reduce the computational cost related to collision detection during the evaluation of an agent, as well as facilitate the quantitative comparison of different morphologies (see section 3). Structural units are the basic building blocks from which the agent's morphology is constructed. 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 structural 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, each of which has the default radius. Note that the volume is thus not conserved across the split event: whether volume concentration in a necessary component of morphogenesis remains to be tested. Although the agent grows through repeated division of units, and units retain a single genome which regulates the diffusion of a constellation of gene products, the structural units used in this model are not to be equated with the biological concept of a cell, such as in the AES system [5], nor are they equivalent to the structural units employed in the parametric models reported in [21, 23, 2, 16]. 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.

Each structural unit contains: zero to six joints attaching it to 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, or another diffusion site within the same unit or in other units. Each of the components of a structural unit are described in more detail in the following sub-sections.

A newly-created unit is attached to its parent unit in one of six possible directions, and is connected by a rigid connector that maintains a constant distance between the units, even though one or both of the attached units may continue to grow. The placement of the new unit is determined by the maximum concentration of the growth gene product found among the six diffusion sites in the parent unit. 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

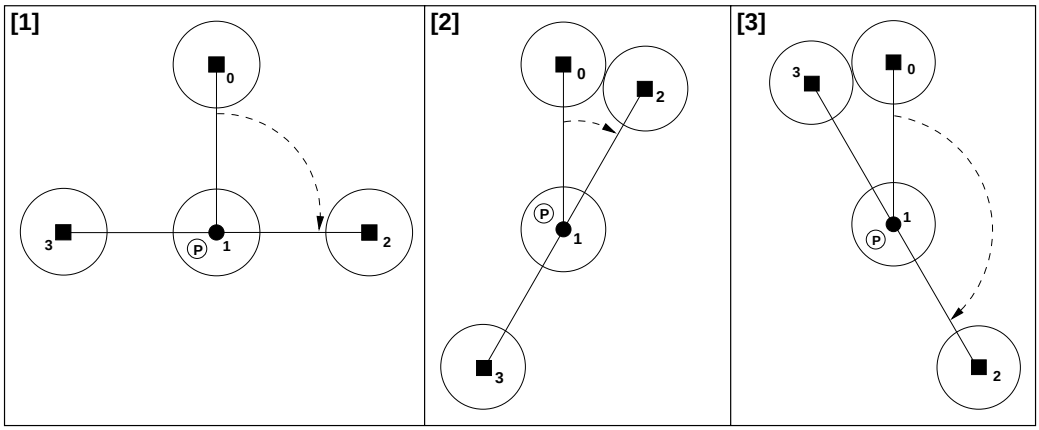



Figure 1: **Architecture of articulated joints** Panels [1] through [3] depict parts of an agent's morphology. In this hypothetical scenario, unit 1 split from unit 0, and units 2 and 3 split from unit 1 during the growth phase. The black squares represent fused joints; the black circles represent rotational 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 configuation 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 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.

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 unit in which the joint resides, and the first unit that split from the unit in which the joint resides. Fig. 1 illustrates the creation and actuation of an agent's joints in more detail. In this way, the final agent is an articulated body composed of connected units; the agent cannot contain cyclically connected units. Fig. 2 depicts, schematically, the morphologies of eight evolved agents.

In addition to the morphology of the agent, neural structure may grow within the developing agent. The growth of neural structure is covered in more detail in the next sub-section. 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 $1$.

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 structural 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).

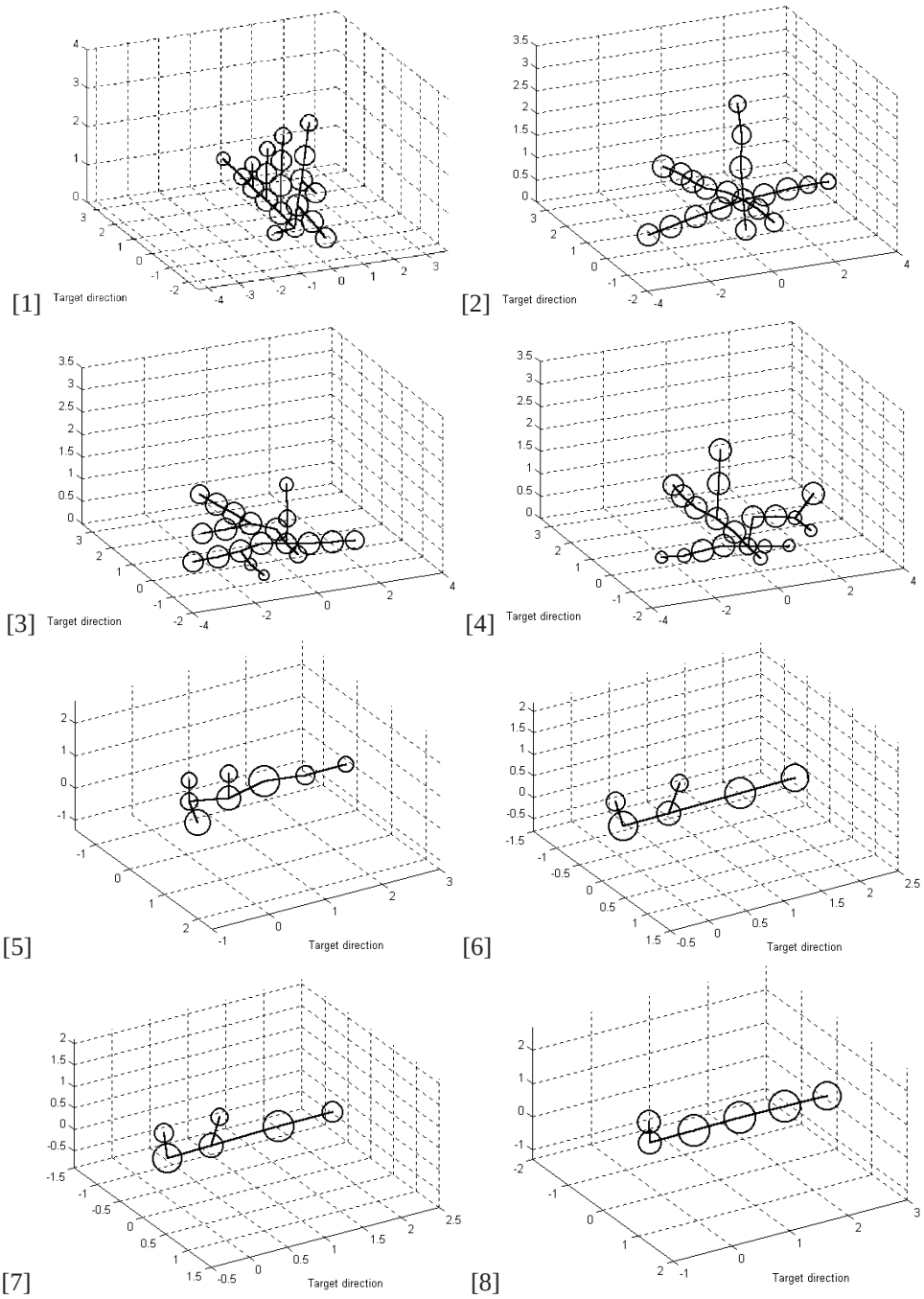



Figure 2: **Sample morphologies of evolved agents** Panels [1] through [4] indicate the fittest agent evolved in four independent evolutionary runs in which the fitness function was to grow as many structual units as possible during the growth period. Panels [5] through [8] indicate the resulting morphologies of the fittest agents in evolutionary runs in which the task was to move as far as possible in a direction indicated by a target object in the environment. The direction of the target (not shown in this figure) lies to the right of the axes.

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 can achieve 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. 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.

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. These additional neuron types allow the agent to perform actions even when deprived of any sensory input.

The different types of neurons are connected to each other by synapses. During development, neurons can migrate between different units. If the neurons are connected by a synapse, this may lead to signal propagation between neurons in distant units.

## 2.2 Differential Gene Expression

For the results reported in the next section, a variable length, floating-point valued genetic algorithm was used. Unlike the recursive parametric encoding schemes used by Sims [21] and Lipson [16], each genome is treated as a genetic regulatory network [11, 5, 20], in which genes produce gene products that either have a direct phenotypic effect or regulate the expression of other genes.

The initial, random genomes in the genetic algorithm population have a length of $100$ floating-point values; each value is rounded to a precision of two decimal places. Values range between $0.00$ and $1.00$. For each genome to be evaluated in the population, it is first copied into the single structural 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 region 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 region indicators.

During the growth phase, the genes may emit gene products: gene products are treated as chemicals which diffuse out from the site of gene expression, and spread to neighbouring diffusion sites, and to a lesser degree, into neighbouring structural units.

Fig. 3 provides a pictorial representation of a genome directing the growth of an agent. The seven floating-point values following a gene's promotor region supply the parameter values for the gene. If the first value ($P1$ in Fig. 3) 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. 3) indicates which of the $24$ possible gene products regulates the gene's expression. The third value ($P3$ in Fig. 3) indicates which of the $24$ possible gene products is produced if this gene is expressed. The fourth value ($P4$ in Fig. 3) 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. 3) 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. 3) denote the concentration range 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 single structual unit is injected with a small amount of gene product at gene product diffusion site $1$. During the initial time steps of development, the gene product diffuses to the neighbouring four diffusion sites, and thence into the diffusion site diametrically opposite to site $1$. In this way we establish a diffusion gradient within the unit, analagous to the establishment of a gradient of maternal gene product in fruit flies, which leads to the determination of the primary body axis within these organisms [1]. Indeed the majority of body plans reported in this paper have a clearly visible primary body axis, and a high degree of bilateral symmetry (see Figs. 2, 4 and 9).

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 gene products. There are $24$ different types of gene products. Two affect the growth of the unit in which they diffuse. At each time step of the development phase, the difference between the concentration of these two gene products is computed. If the difference is positive, the radius of the structural unit is increased a small increment; if the difference is negative, the unit does not grow in size. Thus these two gene products function as growth enhancer and growth repressor, respectively. If the radius of a structural 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.

There are then $17$ other gene products 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 ex-

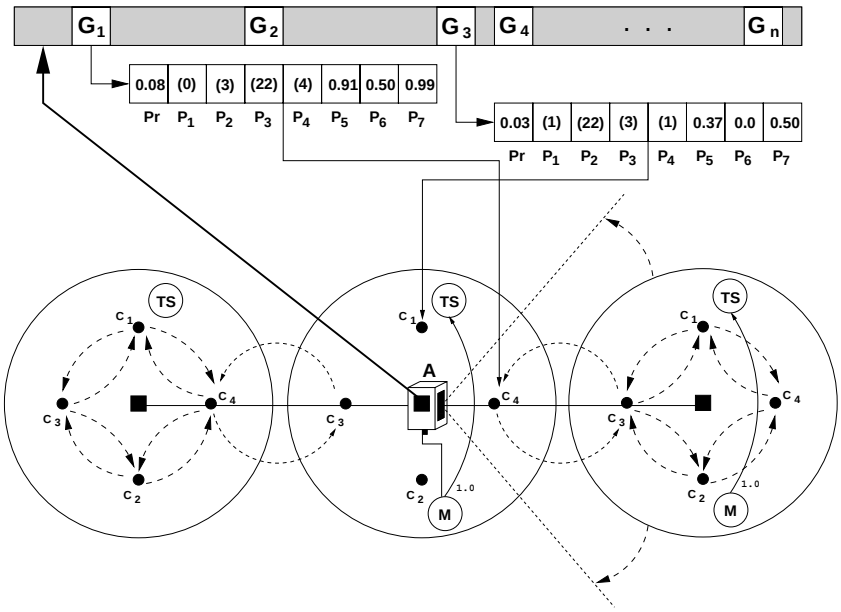



Figure 3: **Ontogenetic interactions in a developing agent** A schematic representation, from the side, of three structual units of an agent are shown above. Four of the six gene product diffusion sites are shown: the other two lie at the top and bottom of the spherical units. The genome of the agent 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 (diffusion in the second unit is not shown for clarity). All three units contain a touch sensor neuron, ($TS$) but the motor neuron ($M$) in the first unit was deleted by gene product diffusion. In the other two units, the touch sensor and motor neurons are attached by an excitatory synapse. Note that gene products have inverted the direction of the two synapses. If the motor neurons do not acquire any new input synapses from sensors during growth, these two units will contain non-actuated joints during the evaluation phase. The second unit contains an actuator ($A$), innervated by the resident motor neuron. If the actuator receives motor commands during the evalution phase, the third unit will rotate relative to the other two units.

pression 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

Based on the changing concentrations of growth enhancing and growth repressing gene products during development, continuous growth from a single structual unit into a three-dimensional, multi-unit agent is achieved, as described in the previous sub-section. Six frames from a typical growth pattern are depicted schematically in Fig. 4. Cellular encoding [7] has been incorporated into our model to achieve the correlated growth of morphology and neural structure in a developing agent. Cellular encoding is a method for evolving both the architecture and synaptic weights of a neural network by starting with a simple neural network, and iteratively or recursively applying a set of graph rewrite rules to transform the simple network into a more complex network. In AO, high concentrations of certain gene products can trigger graph rewriting rules that modify or increase the amount of neural structure in a structural unit. In this way, both morphology and neural structure can change together during the growth phase. This stands in contrast to the neural development model used by Delleart and Beer [4], in which innervation of cells occurs only after cell division has ceased. 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 structural 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. 3). When a structural unit undergoes a split event, any neurons at the diffusion site where the split event was initiated are moved to the neighbouring diffusion site in the new unit. For example, if a structural unit splits, and the new unit is attached to 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 [10]—it does allow for neurons in distant units to remain connected. Any one agent may contain up to 100 neurons, and 100 synapses: if, during growth, either of these maximii are reached, any subsequent graph rewrite rules that attempt to add neural structure are ignored.

Each of the $17$ gene products responsible for neural development correspond to one operation which 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 operations are summarized in Table 1. A pictorial representation of the first two rewrite rules are shown in Fig. 5. Rewrite rules 11 to 14 listed in Table 1 change the synapse or neuron pointer.

If the concentration of one of these $17$ gene products at a diffusion site exceeds a concentration of $0.8$, and there is neural structure at that site, the corresponding rewrite rule is applied to the neural structure there. Using this neural development scheme, the

| Gene Product | Cellular Encoding Operation Description |
|---|---|
| 0 | Split the current neuron into two neurons. Move the output synapses of the original neuron to the new neuron. Connect the original neuron to the new neuron with a synapse of positive weight. |
| 1 | Split the current neuron into two neurons. Copy the input and output synapses to the new neuron. Connect the two neurons to each other using two synapses of positive weight. |
| 2 | Move the current neuron to the previous diffusion site. |
| 3 | Move the current neuron to the next diffusion site. |
| 4 | Move the head of the current synapse to the current neuron. |
| 5 | Move the tail of the current synapse to the current neuron. |
| 6 | Increment the weight of the current synapse by $0.01$. |
| 7 | Decrement the weight of the current synapse by $0.01$. |
| 8 | Duplicate the current synapse. |
| 9 | Delete the current neuron, including any ingoing and outgoing synapses. |
| 10 | Delete the current synapse. |
| 11 | Move the neuron pointer to the next neuron at the current diffusion site. |
| 12 | Move the neuron pointer to the previous neuron at the current diffusion site. |
| 13 | Move the synapse pointer to the next synapse at the current diffusion site. |
| 14 | Move the synapse pointer to the previous synapse at the current diffusion site. |
| 15 | Change the type of the current neuron (Touch sensor → Proprioceptive sensor → Light sensor → Bias neuron → Oscillatory neuron → Internal neuron → Motor neuron → Touch sensor) |
| 16 | Change the type of the current neuron (Motor neuron → Internal neuron → Oscillatory neuron → Bias neuron → Light sensor → Proprioceptive sensor → Touch sensor → Motor neuron) |

Table 1: **Phenotypic effect of neural development gene products** The rewrite rules are triggered when the gene product responsible for that rule reaches a concentration threshold. When a rule is triggered at a diffusion site, the rule is applied to the current neuron or synapse at that site. In the current model, rules cannot be applied to more than one neuron or synapse at the same time.

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. Growth of both morphology and neural structure halts when 300 time steps have elapsed.

# 3 Results and Analysis

The results reported below were collected from evolutionary runs lasting $200$ generations and using a population size of $200$. The mutation rate was tuned to produce an average of one mutation per genome copied. Unequal crossover was used, which allows for gene duplication and deletion, as well as the production of child genomes which are not nec-

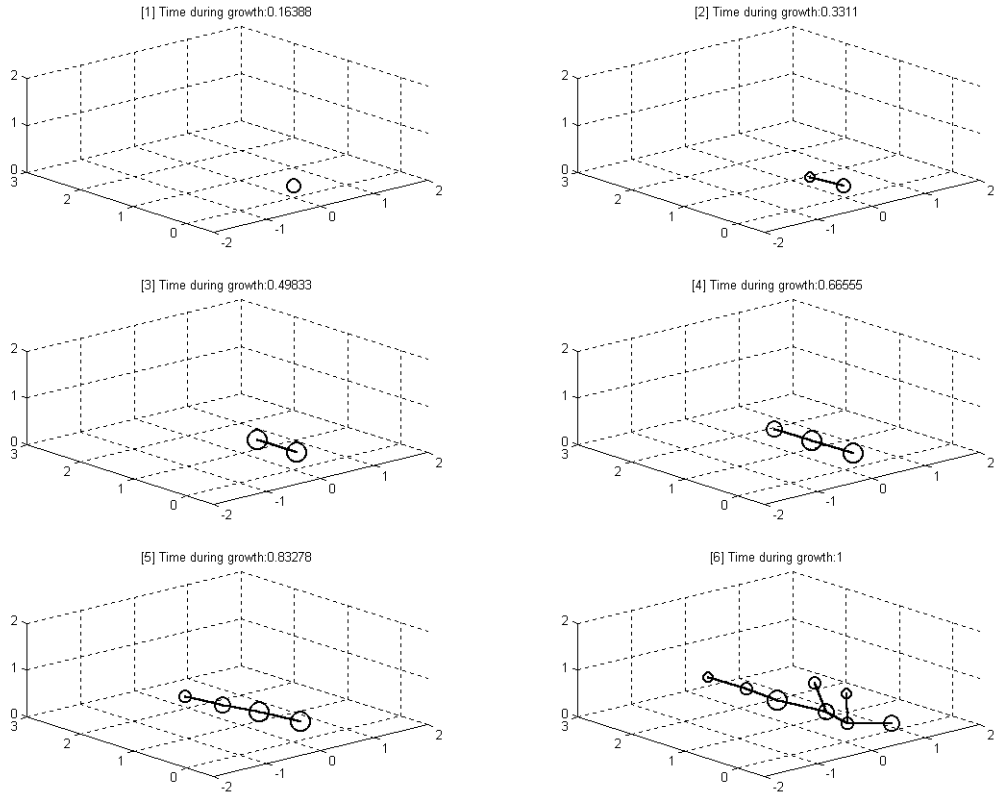



Figure 4: **Progression of growth in an evolved agent** Panels [1] through [6] indicate the progression of growth of an evolved agent. The nodes are scaled to half their actual size for clarity.

essarily the same length as the parental genomes.[2] Crossover was accomplished using tournament selection, with a tournament size of 2.

Each genome in the population is evaluated as follows: The genome is copied into a single structural unit, which is then placed in a virtual, three-dimensional environment. A target object is placed 20 units[3] to the north of the unit. Morphological and neural development is allowed to proceed, as described in the previous section, for 300 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. A small amount of noise is introduced into the environment by applying random instantaneous external forces, of varying but mild magnitudes, to the structural units of the agent during the evaluation period. At the end of this evaluation period, the northern distance from the origin to the agent's trailing, southernmost structural unit is recorded. The agent is then regrown and re-evaluated nine more times. The northern distances achieved are averaged and recorded as the fitness of the agent grown from the given genome. By averaging the agent's fitness

[2]Unequal crossover involves choosing crossover points in the two parent genomes that may be located as different positions along the two genomes. In the standard crossover operation, the crossover points are chosen at an equal distance along the two genomes.

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

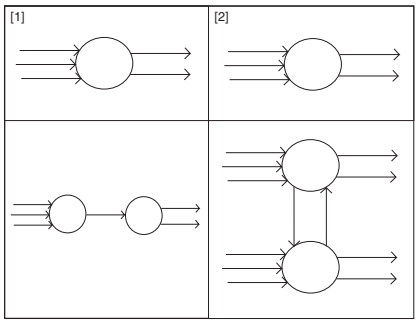



Figure 5: **Two cellular encoding rewrite rules** Panel [1] illustrates the result of a serial divide event, initiated by high concentrations of gene product 0. Panel [2] illustrates the result of a parallel divide event, initiated by high concentrations of gene product 1. (See Table 1).

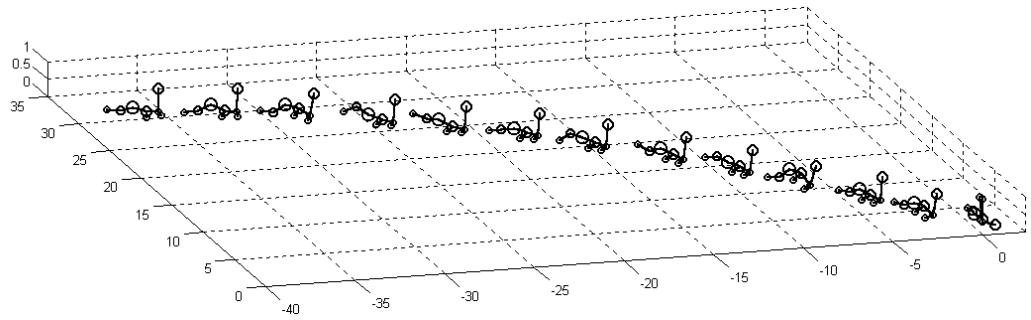



Figure 6: **Behaviour of an evolved agent** A schematic representation of the movement of the agent depicted in Fig. 4. Note that the agent traces a curved, rather than a direct path towards the target direction.

values, we avoid evolving agents that only perform well in a particular noisy environment. In other words, selection favours those agents which are stable against fluctuations in the external environment.

The task is thus to move as far as possible in a northerly direction—indicated by a target object—in a noisy environment. Agents $[5]$ through $[8]$ in Fig. 2 report the morphologies of the fittest agents produced by four independent evolutionary runs. Figs. 4 and 6 illustrate the growth, and subsequent locomotion, of agent $[5]$ from Fig. 2.

A total of $10$ independent evolutionary runs were conducted, and the genotypes of the fittest agents during the runs were recorded. This produced a series of genotypes $g_{i_1}^{i_2}, g_{i_2}^{i_3}, \ldots g_{i_{k-1}}^{i_k}$, where the agent grown from genotype $g_{i_1}^{i_2}$ was the fittest agent in the population for generations $i_1$ to $i_2$. Thus, genotype $g_{i_2}^{i_3}$ appeared in the population in generation $i_2$, and produced an agent fitter than the agent grown from genotype $g_{i_1}^{i_2}$.

The morphological difference $m(t)_i^j$ between two agents $a_i$ and $a_j$, composed of $n_i$ and $n_j$ structural units respectively, at time step $t$ during the growth phase was computed using

$$m(t)_i^j = \frac{\sum_{k=1}^{N}(|\mathbf{p}(t)_k^i - \mathbf{p}(t)_k^j| + |r(t)_k^i - r(t)_k^j|)}{N},$$

where $N = \min(n_i, n_j)$. $\mathbf{p}(t)_k^i$ and $\mathbf{p}(t)_k^j$ give the three-dimensional positions of structural

units $k$ of agent $i$ and agent $j$, respectively, at time step $t$ during the growth period. Thus $|\mathbf{p}(t)^i_k - \mathbf{p}(t)^j_k|$ gives the distance between units $k$ of agents $i$ and $j$ at time step $t$. $r(t)^i_k$ and $r(t)^j_k$ give the radii of structural units $k$ of agents $i$ and $j$ at time step $t$. Thus $|r(t)^i_k - r(t)^j_k|$ gives the difference between the radii of the two units. Note that this measure does not capture differences in agent pairs in which one agent is much larger than the other. However, due to the growth mechanisms involved, agents of differing size usually show pronounced differences among the first $N$ units.

By recording the morphologies of agents $i$ and $j$ for each time step during their development, and by translating their positions so that the positions of their originating structual units are identical, we can compute $m(t)^j_i$ for each time step of their development. For time steps in which growth is identical, $m(t)^j_i = 0$. However, if their ontogenetic development diverges, their morphologies may differ, and $m(t)^j_i > 0$. Because there is only noise during phenotype evaluation, i.e. after growth has been completed, any phenotypic difference must be caused by genetic change. Thus we can then record time step $t$ as the point during development when a mutation in agent $j$ led to a change in its development. As in the results reported here, agent $j$ replaced agent $i$ as the fittest agent in the population, so we can conclude with reasonable certainty that the mutation which led to ontogenetic divergence of this pair was a beneficial mutation.[4]

It was found that for the majority of the agent pairs compared, morphological divergence began at $t_0$. However, there were many agent pairs in which morphological divergence occurred later in development. Fig. 7 reports those agent pairs in which morphological divergence occurred later than $t_0$. The time of morphological divergence, given by time $t$ in $m(t)^j_i$, is scaled between $0$ and $1$, with $1$ representing a divergence during the last time step of the growth period. The time of morphological divergence is plotted against $m(t_e)^j_i$, where $t_e$ is the final time step of the growth phase.

Each of the agents extracted from the $10$ evolutionary runs was then evaluated $30$ times in a noisy environment, to better determine its fitness. For each pair of agents $i$ and $j$, the absolute difference between their averaged fitness values was calculated, and is referred to as the behavioural difference of this agent pair. The behavioural difference of each agent pair is plotted against that pair's $m(t)$ value in Fig. 8.

Using the same parameters given above, a second fitness function was implemented in which agents must push a large square block in a specific direction. The length of each side of the block is $70$ times that of the starting radii of structural units. In this way it is possible to exert indirect selective pressure towards larger agents. The maximum number of units that could be used to construct an agent was increased to $50$. Fig. 9 depicts the fittest agents extracted from two independent evolutionary runs; both agents contain the maximum possible number of structural units. The first agent appeared in its population after $83$ generations; the second agent appeared in its population after $51$ generations. The shadings of the structural units denote the architecture of the neural structure embedded within them, and are explained in more detail in the figure caption. Both agents use an anterior appendage to push against the block, and establish an oscillatory motion in the posterior appendage to achieve slow forward motion (the anterior to posterior axes

[4]This assumes that the mutation causing ontogenetic divergence is not overshadowed by a second mutation appearing in agent $j$ which is actually the beneficial mutation, an event which has a very low probability of occurring.

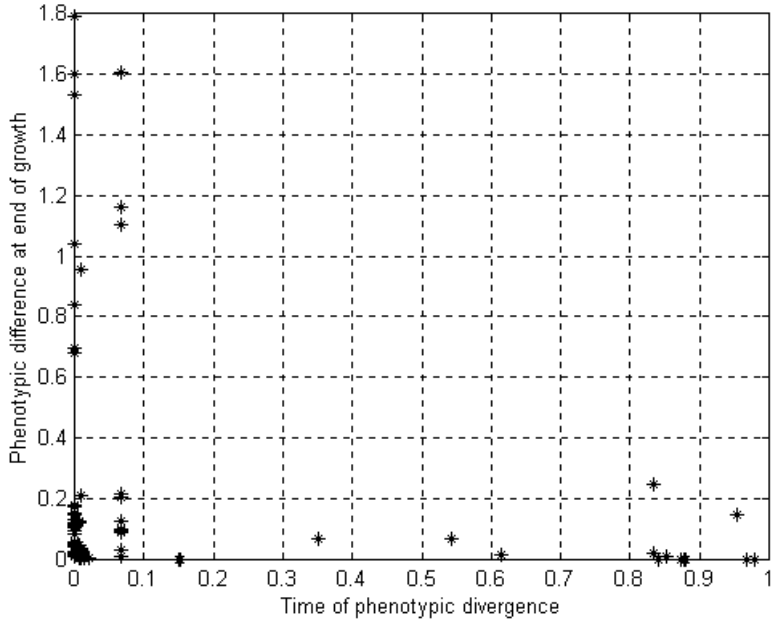



Figure 7: **Time of morphological divergence versus magnitude of final morphological divergence**

of the agents are shown from left to right in Fig. 9). The agent in panel [2] uses two appendages to push the block. The lower appendage contains all white structural units, except for the end unit. The upper appendage rests on the lower appendage, and contains two neighbouring black units. This appendage provides support for the pushing action of the lower appendage; however, the upper appendage also pushes against the block during part of the evaluation period.

# 4 Discussion

The four sample morphologies shown in panels [5] to [8] in Fig. 2 seem quite similar, and indeed the behaviours of these agents are also very similar. For the task of directed locomotion in a noisy environment, a long primary body axis tends to evolve in the direction of the target object, with a few outgrowths at the posterior end of the agent. Motion then proceeds by oscillations along the body axis, produced by propagation of signals from touch sensors to motor neurons. Orientation is maintained using the outgrowths, which keep the agent stable while its oscillations propel it towards the target object. In most evolutionary runs, there is little or no modification of the original, embryonic neural networks implanted in the growing agents. Note that the number of structural units embodied in these agents was not a criterion in the fitness function, but rather is an indirect result of the selection pressure specific to this task.

In contrast, however, if agents are evolved to incorporate a maximum of possible structural units, morphologies evolve in which the differences are much more pronounced (see panels [1] to [4] in Fig. 2). Similarly, the morphologies of the two block pushing

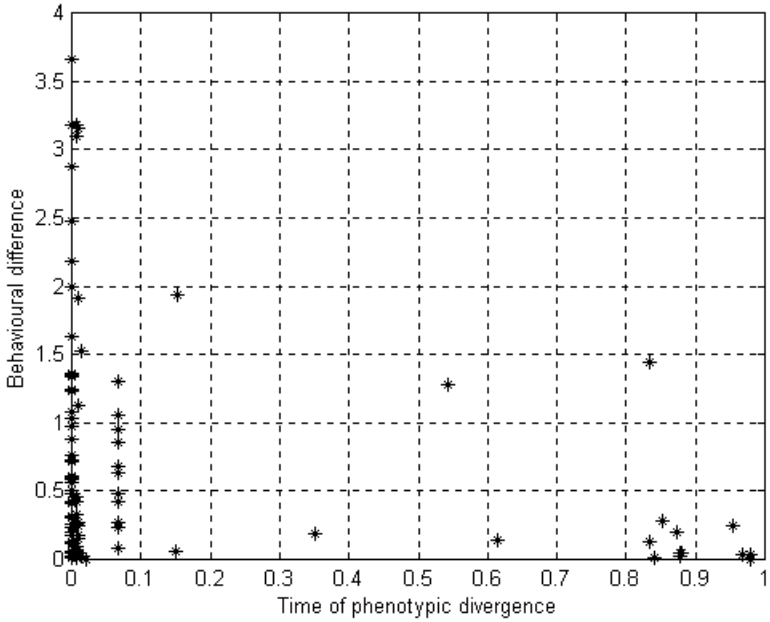


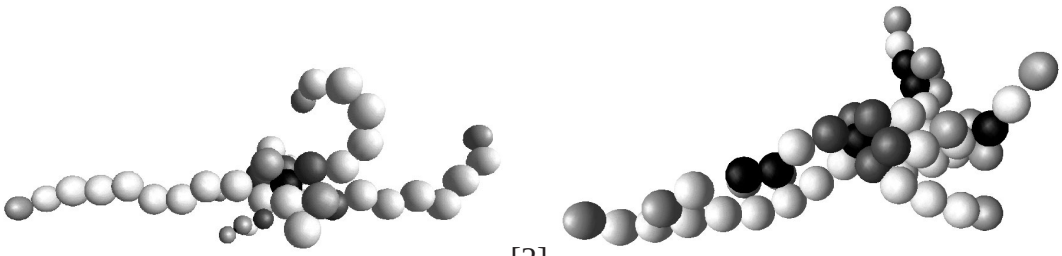

Figure 8: **Time of morphological divergence versus behavioural difference**

agents exhibit differing morphologies.

This indicates that the repeated convergence on a similar body plan in the directed locomotion task is specific to the task, and not a universal trait of the model introduced here. What the results suggest is that in the task in which the agent must make use of its internal neural structure, modification of the agent's morphology is sufficient to produce an agent that performs well at the task. Thus, the resulting morphologies for the directed locomotion task work well with the user-encoded neural network. We plan in future studies to begin agent development with a single neuron in the originating structural unit of the agent, and determine whether more varied phenotypes evolve.

Fig. 7 indicates that for agent pairs in which a mutation causes morphological divergence during ontogeny, the final morphological difference between the pairs shows more variance for divergences that occur early during ontogeny. For agent pairs in which the morphological divergence has a late ontogenetic onset (indicated by points further to the right in the figure), the final morphological difference is not as pronounced.

Although this result is somewhat intuitive, we see a similar pattern in Fig. 8, in which the fitness improvement between agent pairs has a higher variance in those pairs in which the mutation separating them has an early ontogenetic onset. This suggests a correlation between the morphological and behavioural difference of an agent pair. Although no such correlation was detected when behavioural difference was plotted against morphological difference (not shown here), we hypothesize that a more sophisticated measure of behavioural difference may reveal a correlation. Note that the behavioural difference only captures the difference between how well two agents perform; it does not measure the different ways two agents may perform the same task. Thus, two agents which travel, on average, the same distance during evaluation, but travel that distance in different ways,



[1] [2]

Figure 9: **Two agents evolved for block pushing** The fittest agents extracted from two independent evolutionary runs. The block is not shown in the figure for the sake of clarity, but lies just to the left of both agents. 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 grey units indicate the presence of sensor neurons, but no motor neurons. The black units indicate there are neither sensor nor motor neurons within that unit.

would have a very low behavioural difference. A more sophisticated measure would be required to distinguish between different methods of locomotion, and will be the focus of future studies.

Figs. 7 and 8 suggest that mutations with an earlier ontogenetic onset have a more variable morphological and behavioural effect than mutations with a later ontogenetic onset. Because this was shown for agent pairs in which the agent bearing the mutation was fitter than the other agent, it follows that this holds for selectively advantageous mutations. In future we plan to test whether this correlation also holds for deleterious mutations. We hope to show in future work that this mechanism proves useful during incremental evolution, in which a population of agents are evolved to solve a relatively simple task, and are then presented with a slightly more challenging task. We hypothesize that the combination of ontogenetic development and differential gene expression allows evolution to continually explore the altered fitness landscape by exploiting beneficial mutations with differing times of ontogenetic onset, and thus differing magnitudes of phenotypic effect. That is, mutations with varying magnitudes of phenotypic effect are always available during search. In the case of parametric encoding schemes, it is not always obvious whether mutations of a given magnitude of phenotypic effect are possible, which will allow artifical evolution to escape local optima and continue exploration of the search space. We predict that this property of the AO system makes it more evolvable than parametric encoding schemes.

By evolving agents to push a block much larger than their constituent structural units, we were able to indirectly evolve agents with a large number of units. Moreoever, we find evidence of functional specialization in these agents. In Fig. 9, we note that in the central part of the agent, several units lose their motor capabilities during growth. This is a result of the migration of motor neurons out of these units during ontogeny, but it is interesting to note that the sensor neurons do not migrate. This suggests that the sensory capabilities of this central part of the agent may serve a purpose, perhaps by activating motor neurons in distant structural units. In the agent depicted in panel [2] of Fig. 9, we note that the appendages of the agent are composed of differentiated units (indicated by

the two pairs of neighbouring black units), unlike the appendages of the agent in panel [1]. This suggests that the evolution of differentiated, repeated structure has occurred in this second population.

# 5 Conclusions and Future Research

In this report we have introduced the Artificial Ontogeny system, which uses genetic regulatory networks to grow a complete agent from a single building block. It was demonstrated that the AO system can evolve functioning agents in a physically-realistic virtual environment: specifically, agents were evolved to perform directed locomotion and block pushing in a noisy environment. Although agents for the directed locomotion task tended to converge on a similar body plan, agents evolved for large size, and for block pushing demonstrated a wide range of morphologies during independent evolutionary runs.

Also, we have shown that beneficial mutations which have an early ontogenetic onset lead to varying magnitudes of morphological differences between the agent without the mutation and the agent in which the mutation first appears. Moreover, it was demonstrated for some agent pairs, the agent containing the mutation tends to achieve a much higher performance increase over its predecessor, compared with agent pairs separated by a mutation which has a later ontogenetic onset. This property of the AO system demonstrates that in this model, mutations of varying phenotypic effects are always available during search, which may make the AO system more evolvable than parametric encoding schemes. This hypothesis will be explored in future studies.

We have also shown that for agents composed of many units, functional specialization tends to evolve in these populations. This was indicated by the repeated appearance of agents with a central structure that had lost its motor capabilities, but not its sensory capabilities. Moreover, in one population, an agent with repeated differentiated structure was found, suggesting that a recursive, parametric encoding scheme is not necessary for achieving such structure. In future studies we plan to better elucidate the connection between scaling the number of possible units composing an agent, functional specialization, and the appearance of repeated, differentiated structure. Repeated, differentiated structure is desirable if we wish to transfer evolved designs in an efficient manner onto real-world modular robots.

Finally, we have begun to analyze the resulting gene expression patterns from the evolved agents. We hope to shed some light on how artificial evolution alters regulatory networks in order to achieve fitter phenotypes. We also hope to find evidence of the evolution of regulatory genes that control suites of structural genes, analagous to the homeotic genes found in biological organisms [6]. It has been demonstrated in natural evolution that homeotic genes are closely coupled with the modular property of body plans [19]. This suggests that the appearance of homeotic genes in artificial evolution could facilitate the transfer of evolved agents to modular robots: adaptive changes to the evolving robot body plan may require only the rearrangement of functional units composed of many building blocks, as opposed to low-level changes to all or some of the building blocks comprising the robot.

# References


[1] Anderson, K. V. & C. Nüsslein-Volhard (1984) “Information for the Dorso-Ventral Pattern of the *Drosophila* Embryo is Stored in Maternal mRNA.”, Nature, Vol. 311, pp. 223–227.

[2] Bongard, J. C. & C. Paul (2000) “Investigating Morphological Symmetry and Locomotive Efficiency Using Virtual Embodied Evolution”, Proceedings of the Sixth International Conference on Simulation of Adaptive Behaviour, MIT Press, pp. 420–429.

[3] Brooks, R. A. (1990) “Elephants Don’t Play Chess”, Robotics and Autonomous Systems, Vol. 6, pp. 3–15.

[4] Delleart, F. & R. D. Beer (1994) “A Developmental Model for the Evolution of Complete Autonomous Agents”, Artificial Life IV, MIT Press, pp. 246–257.

[5] Eggenberger. P. (1997) “Evolving morphologies of simulated 3D organisms based on differential gene expression”, Proceedings of the Fourth European Conference on Artificial Life, Springer-Verlag, pp. 205–213.

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

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

[8] Hara, F. & R. Pfeifer (2000) “On the Relation among Morphology, Material and Control in Morpho-Functional Machines”, Proceedings of the Sixth International Conference on Simulation of Adaptive Behaviour, MIT Press, pp. 33–40.

[9] Ijspeert, A. J. & J. Kodjabachian (1999) “Evolution and Development of a Central Pattern Generator for the Swimming of a Lamprey”, Artificial Life **5**:3, pp. 247–269.

[10] Kater, S. B. & P. B. Guthrie (1990) “Neuronal growth cone as an integrator of complex environmental information”, Cold Spring Harbor Symposia on Quantitative Biology, Volume LV, Cold Spring Harbor Laboratory Press, pp. 359–370.

[11] Kauffman, S. A. (1993) *The Origins of Order*, Oxford University Press.

[12] Kawai, N. & F. Hara (1998) “Formation of Morphology and Morpho-Function in a Linear Cluster Robotic System.”, Proceedings of the Fifth International Conference on Simulation of Adaptive Behaviour, MIT Press, pp. 459–464.

[13] Kirschner, M. & J. Gerhart (1998) “Evolvability”, Proc. Nat. Acad. Sci, 95:8420-8427.

[14] Komosinski, M. & S. Ulatowski (1998) “Framsticks: Towards a Simulation of a Nature-Like World, Creatures and Evolution”, Proceedings of 5th European Conference on Artificial Life, Springer-Verlag, pp. 261–265.

[15] Lichtensteiger, L. & P. Eggenberger (1999) “Evolving the Morphology of a Compound Eye on a Robot”, Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot ’99), pp. 127–134.

[16] Lipson, H. & J. B. Pollack (2000) “Automatic Design and Manufacture of Artificial Lifeforms”, Nature 406, pp. 974–978.

[17] McGeer, T. (1990) “Passive Dynamic Walking”, International Journal of Robotics Research, Vol. 9, pp. 62–82.

[18] Pfeifer, R. (1999) *Understanding Intelligence*, MIT Press.

[19] Raff, R. A. (1996) *The Shape of Life: Genes, Development, and the Evolution of Animal Form*, University of Chicago Press, pp. 321–361.

[20] Reil. T. (1999) “Dynamics of Gene Expression in an Artificial Genome–Implications for Biological and Artificial Ontogeny”, Proceedings of the Fourth European Conference on Artificial Life, Springer-Verlag, pp. 457–466.

[21] Sims, K. (1994) “Evolving 3D Morphology and Behaviour by Competition” Artificial Life IV, MIT Press, pp. 28–39.

[22] Terzopoulos, D., T. Rabie & R. Grzeszczuk (1996) “Perception and learning in artificial animals”, Artificial Life V, MIT Press, pp. 313–320.

[23] Ventrella, J. (1994) “Explorations of Morphology and Locomotion Behaviour in Animated Characters”, Artificial Life IV, MIT Press, pp. 436–441.

[24] Wagner, G., & L. Altenberg (1996) “Perspective: Complex adaptations and the evolution of evolvability”, Evolution, 50:(3), pp. 967–976.