Evolution of Vision Capabilities in Embodied Virtual Creatures

Marcin L. Pilat[*]
Department of Computer Science
University of Calgary
2500 University Dr. NW
Calgary, AB, T2N 1N4, Canada
mpilat@ucalgary.ca

Christian Jacob
Department of Computer Science
University of Calgary
2500 University Dr. NW
Calgary, AB, T2N 1N4, Canada
cjacob@ucalgary.ca

ABSTRACT

We evolve light following behaviours in virtual creatures through neural network training using an incremental evolution approach. The neural controllers of creatures evolved for movement are augmented with simple visual neurons and neural connections. Using an evolutionary algorithm, the resulting creatures are trained to identify and follow a light source. Through this process, we are able to train the neural controllers to create various light following behaviours. Many of the evolved behaviours show stability and adaptiveness to environmental perturbations of body orientation.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.2.6 [**Artificial Intelligence**]: Learning—*Connectionism and neural nets*; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding

General Terms

Experimentation

Keywords

Evolution, Learning, Artificial Life, Virtual Creatures, Genetic Algorithms, Vision

1. INTRODUCTION

The ability to sense the environment is crucial for life. Organisms have evolved an assortment of different sensing capabilities in order to find food or avoid predators. In many animal species, these sensory systems are also used for communication, enabling social interactions, group dynamics, and emergence of higher level biological phenomena.

The interplay between physical structures (morphologies) and sensory elements is a crucial component of an animal's survival. Virtual creatures act as model virtual animals (or animats) and offer an ideal test-bed to study the interactions between morphologies and sensing models. As Wake argues in [13], the study of virtual morphologies and development will provide clues to answer many open questions in biology, such as the existence of different body types and effects of perturbation on natural ecosystems.

In this paper, we present a simple spherical sensory model for virtual creatures which resembles vision or hearing systems of animals by locating a source object (e.g., a light source) in the environment. The virtual creature model is an adaptation to Sims' Blockies model [12]. The location of the source object is processed through the neural network of the virtual creature, modifying its behaviour.

In contrast to previous approaches (e.g., [12]), we use incremental evolution to reuse previously evolved morphologies and neural controllers, and provide means of comparison with experiments in direct evolution of walking and sensing creatures. The virtual creature morphologies and controllers are first evolved using a standard genetic algorithm with various locomotion based fitness criteria. The neural network controllers of selected resulting virtual creatures are then infused with random sensing networks and are further evolved using the task of light following.

2. BACKGROUND

The pioneering work on the evolution of morphologies and controllers of virtual creatures was performed by Sims [11, 12]. Sims used a genetic algorithm to evolve both the morphologies and controllers of articulated virtual creatures called Blockies. His graph-based generative encoding produced modularity and symmetry in the morphologies. Neural network controllers were embedded into the creature modules. The results showed various behaviours for independently evaluated learning tasks, such as swimming, walking, jumping, and following a light source.

Evolution of vision capabilities has been studied both in virtual creatures and in robotics. Sims in [12] has used photo sensors to evolve light following for creatures in land and aquatic environments. The sensors provided the neural network controller with a relative direction to the light source. His fitness function measured the average creature speed toward the light over several trials using different light source locations. In [11], Sims modified the photo sensors to sense

---

[*] The author is currently working at Defence R&D Canada, Centre for Operational Research and Analysis, Ottawa, ON, Canada.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'10, July 7–11, 2010, Portland, Oregon, USA.
Copyright 2010 ACM 978-1-4503-0072-8/10/07 ...$10.00.

other virtual creatures and a box, enabling the possibility for a competitive box-grabbing task.

Miconi et al. [7] have successfully evolved virtual creatures for the tasks of locomotion and box-grabbing. In contrast to Sims' work, their model used standard McCulloch-Pitts neurons. External sensors provided the neural networks with information on the distance to the opponent and distance to the box. Chaumont et al. [3] has experimented with a modified Sims model to evolve virtual catapults that were able to throw a ball over large distances.

Touch sensors have been used in [3, 6, 11] to detect contacts between body parts of two creatures or the environment. However, Chaumont et al. conclude in [3] that little research has been done in virtual creature sensing and sensor optimization. Sensing capabilities are studied intensively in mobile robotics [14], especially in contexts requiring the co-operation of several robots (e.g., Dorigo's swarm-robotics approach [5]).

The incremental evolution approach stems from robotic applications where morphologies and some behaviours (such as walking) can be engineered into the system, and additional abilities (such as sensing) can then be added on at a later time (e.g., the layered subsumption architecture of Brooks [2] or behaviour chaining of Bongard [1]). Incremental Evolution has been used to evolve robot controllers as in [4]. Nolfi et al. [8] provide an introduction to the evolution of artificial neural networks, which we are using to control our virtual creatures.

3. VIRTUAL CREATURES

We evolved the morphologies and controllers of virtual creatures for the training task of light following. The creatures used simple sensors that enabled them to identify the location of objects in the environment. In this section, we describe the virtual creature model and the sensing model that we use in our experiments.

3.1 Virtual Creature Model

Our virtual creature model was based on the Blockies model introduced by Sims [12]. Sims' model is ideal for studies of virtual creature morphologies since it combines a simple phenotype and a powerful generative genotype representation. The virtual creature model used in this work is covered in greater detail in [10]. Figure 1 provides a sample evolved virtual creature showing its morphological genotype, phenotype, and the neural network.

The creature morphologies were composed of cuboid body parts connected with simple hinge joints, providing each connected body part with one degree of freedom with respect to its parent part. The joints defined arbitrary orthogonal axes of movement and were thus not constrained to movement along a plane. This complicated the learning task by increasing the amount of required sensory information.

We modelled the phenotype of a virtual creature as a rooted tree (Figure 1(b)). Nodes of the tree described body parts of the creature, whereas the links represented the joints (Figure 1(d)). The morphological genotype (Figure 1(a)) had the form of a directed graph with possible cycles and self loops. This graph-based generative encoding stored information used to build the corresponding phenotype through a recursive unravelling of the genotype nodes controlled by a recursion level parameter.

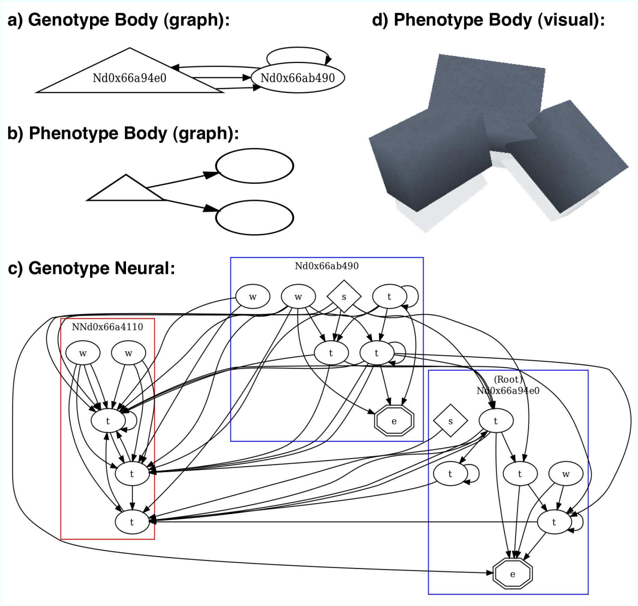


Figure 1: A sample evolved virtual creature showing: a) the genotype of the morphology, b) the corresponding phenotype using a recursion level of 1, c) the genotype neural network, and d) the graphical representation of the phenotype. In the body graphs, body parts are represented as a triangle (main body part) and ovals. In the neural repesentation, subnetworks are composed of various neurons: computational (t), periodic (w), sensor (s), and effector (e). These subnetworks are embedded within the body parts. The left subnetwork (red frame) is the global node.

Each link in the genotype graph specified the position of the contact point on the parent body, the orientation of the child node relative to the parent, and a scale parameter specifying scaling of the child node relative to its parent node. The scaling was used to generate increasingly smaller/larger body parts. Negative scale values denoted reflections about the main axis of the parent node and allowed for the generation of symmetric arrangements.

The control of a virtual creature was accomplished via a recurrent artificial neural network (ANN) of neurons and neural links specifying a directed graph topology (Figure 1(c)). The ANN was composed of subnetworks embedded in the body parts, each with its own sensors, effectors, and other neurons. Connections between subnetworks were only allowed for neighbouring body parts. A global neural network node (not associated with a physical body) facilitated communication between the local neural networks and enabled a form of centralized control.

The return value of every neuron was in the range [-1,1] to standardize the neural output/input values. Computational neurons were simple neurons with a sigmoidal (hyperbolic tangent) transfer function. Special sinusoidal periodic source neurons were used as waveform generators that feed the network with a periodic signal.

Each body part contained a joint angle sensor neuron and an effector neuron. The sensor neuron stored the value of the body's joint angle with respect to its axis. The effector neuron represented the output of the local neural network and translated to the angular velocity along the joint axis.

3.2 Vision Model

We used a simple radius-based source point vision model as illustrated in Figure 2. A single sensor was embedded in the main body part of each virtual creature and registered all objects within its sensing radius r. The sensor only responded to a single light source object but it could also optionally be configured to respond to multiple other objects, including other creatures.

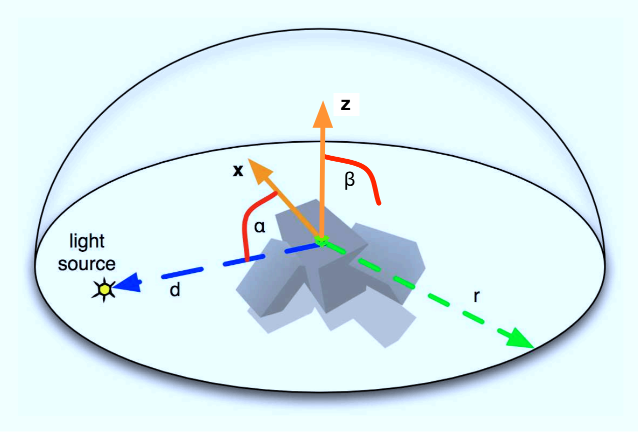


Figure 2: The vision model for light following experiments based on a sensing sphere of radius r and a light source object at a distance d away from the creature. The α angle measures the creature's orientation toward the light along the simulation plane. The β angle measures the creature's orientation perpendicular to the simulation plane.

The most difficult aspect of the sensing model implementation was the method of feeding the sensory information into the neural network of a creature. Enough information needed to be supplied to the ANN while using the least number of neurons and neural connections to limit the genotype size. We identified various methods of feeding the source object location into the neural network.

Source object locations, can be represented via absolute positions, relative vectors, angles, angle signs, and distances. Multiple sensors can also be used for some calculations. Vectors do not require pre-computation and can be fed directly into the ANN. However, the ANN needs to be able to process the vector information. Scalar values, such as angles and distances, need to be precomputed but can be directly used by the ANN with little or no computation.

Since our virtual creature controllers were composed of simple sigmoidal neurons, we supplied the neural network with precomputed scalars instead of vectors. Controllers composed of neurons which can perform complex mathematical function (such as in [11]) would allow for the processing of vector information. However, it is still not clear if offloading the computation from the sensor to the ANN produces better results.

A distance measure provided the neural network with information on the proximity to a source object (shown as d in Figure 2). We scaled the true distance value using a log scale: $\log(d+1) / \log(r)$, where d is the distance value and r the sensing radius. The output value fed to the ANN was inversely proportional to the actual distance and in the range $[0, 1]$. Distance values larger than the sensing radius were given the output value of 1.

Angle measures specified the orientation offset of the virtual creature from a desired orientation (based on a specific metric). The sign of the angle defined the directionality of the angle measure with respect to some fixed axis (i.e., left/right or up/down). Through experimentation, we identified two important angle measures: a measure of the creature orientation along the simulation plane (denoted as α) and a measure of creature orientation perpendicular to the simulation plane (denoted as β). Figure 2 provides a schematic diagram of these measurements.

The α angle measure was calculated using a simulation plane projection of the vector to the source object ($\mathbf{v_p}$) and the vector along the x body axis of the creature ($\mathbf{x_p}$). We defined the angle measure as $\alpha = 1 - \hat{\mathbf{v}}_\mathbf{p} \cdot \hat{\mathbf{x}}_\mathbf{p}$ with a value in the range $[0, 1]$. The alignment of $\mathbf{x_p}$ with $\mathbf{v_p}$ produced an α value of 0. Two symmetric sensor orientations produced to the same α value (with $\mathbf{x_p}$ on different sides of $\mathbf{v_p}$). To distinguish these orientations, we introduced the sign measure $s_\alpha = (\mathbf{v} \times \mathbf{x}).z / |(\mathbf{v} \times \mathbf{x}).z|$ with a value of 1 if $\mathbf{x_p}$ was at or to the left of $\mathbf{v_p}$ and a value of -1 otherwise.

The α angle measure and the sign scalar s_α provided the virtual creature ANN with enough information to process and act on the location of light sources that were relative to the simulation plane. However, flipping the creature's body frame, and correspondingly the sensor and joint axes, reversed its orientation relative to the simulation plane. In flipped configurations, creature actions such as turning right had the opposite effect with respect to the location of the source object.

To combat the flipping problem, we introduced the β angle measure as the cosine of the angle between the z body axis of the creature $\mathbf{z}$ and the simulation plane, defined as $\beta = 1 - \hat{\mathbf{z}} \cdot \hat{\mathbf{z}}_\mathbf{p}$ with the range of $[0, 1]$. The measure had a value of 1 when the projection $\mathbf{z_p}$ did not exist (i.e., $\mathbf{z}$ was perpendicular to the plane). The sign of the β angle measure, calculated as $s_\beta = \mathbf{z}.z / |\mathbf{z}.z|$, indicated whether $\mathbf{z}$ pointed away or toward the simulation plane and enabled the detection of flipped orientations.

4. VIRTUAL CREATURE EVOLUTION

In this section, we describe the evolutionary system and setup that was used to perform the experiments in evolution of light following behaviours of virtual creatures. We introduce the Morphid Academy experimental platform for evolution of functional forms and then present the evolutionary algorithm and the vision infusion process.

4.1 Morphid Academy

Morphid Academy[1] is a virtual laboratory for the evolution of functional forms called Morphids. The abstract concept of a functional form can be best represented using the virtual creature model. However, Morphid Academy can be used to evolve various stationary and mobile physically simulated forms. Detailed information about the Morphid Academy system (called Creature Academy for virtual creature evolution) is provided in [9] and [10].

---

[1] Morphid Academy software can be found on the Morphids Project website at `http://www.morphids.com`.

The Morphid Academy software features physical simulation using the ODE physics engine[2], graphical visualization through the OGRE graphics engine[3], and an ability to perform structured evolutionary experiments using a built-in genetic algorithm. Evolutionary experiments are performed in virtual world instances, called Training Grounds, that can be structured in various topologies.

The hierarchical topology of Morphid Academy experiments realizes an incremental evolution framework. Upon finishing an evolutionary run, Training Grounds can send their best performing individuals to a higher tier Training Ground where they are further improved (via evolution, learning, etc). In this work, we used the Morphid Academy system to streamline the evolution through a 2-tier hierarchical Training Ground approach as shown in Figure 3.

The Tier-1 Training Grounds evolved virtual creatures from initially random populations of creature genotypes for the walking, jumping, and light following tasks. The Tier-2 Training Grounds selected individuals from the results of the Tier-1 walking and jumping experiments, infused the creatures with vision networks, and evolved light following behaviours as described in the following subsections.

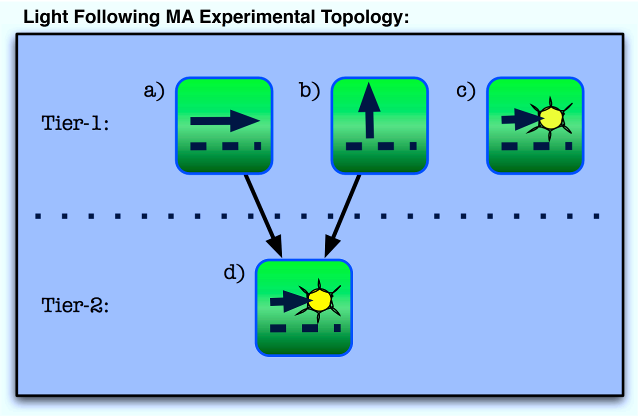


Figure 3: Two-tier Morphid Academy experimental setup for the light following training experiments. The boxes represent Training Grounds running experiments in a) Tier-1 walking, b) Tier-1 jumping, c) Tier-1 light following, and d) Tier-2 light following.

4.2 Evolutionary Algorithm

We utilized a standard steady-state genetic algorithm on a population of creature genotypes with a tournament selection scheme. The simulation environment was composed of a simple simulation surface plane. For the presented work, we disabled creature interactions and only allowed collisions with the ground.

The evaluation of a small set of virtual creatures was performed in parallel in the same simulation environment due to the disabled interactions. The parallel evaluation of 3 tournaments of 3 virtual creatures (9 creatures in total) provided a 5% decrease in simulation time compared to the individual sequential evaluations. The evaluation consisted of: (1) computation through the neural network, (2) actuation of the joints using the values computed by the neural network, and (3) stepping of the physics engine. The integrity of each simulated creature was constantly checked using a computation of joint separation errors to prevent "joint explosions" by disabling invalid creatures.

We used single-objective fitness functions specific to each of the different training tasks. For the locomotion tasks of walking and jumping, a distance-based function computed the displacement of the creature along the simulation plane or the maximum height reached by the main body part of the creature. The fitness for light following experiments was measured using a speed function $\Omega_{sl} = \sum_{i=1}^{N_{iter}} (|\mathbf{v_{i-1}}| - |\mathbf{v_i}|)/N_{iter}$, where N_{iter} is the number of iterations in an evaluation, $\mathbf{v_i}$ is the distance vector from the virtual creature to the light source at time i, and $|\mathbf{v_0}| = 500$.

The Tier-1 experiments used graph-based genetic operators of mutation, crossover, and grafting as defined by Sims [12]. Further information about the specific implementation of the genetic operators for Tier-1 experiments can be found in [9] and [10]. The genetic operator for Tier-2 light following experiments was a topology preserving mutation operator modifying only the weights of neural network connections (with a probability of 0.10 per weight).

4.3 Vision Infusion

The Tier-2 light following experiments studied the problem of adding and evolving light following neural circuitry. For each experiment, we created a population of 300 clones of a randomly selected seed creature from a pool of 41 creatures evolved through Tier-1 locomotion experiments. Each clone was subjected to a vision infusion process which augmented the existing neural networks with vision sensory neurons and neural connections.

A schematic diagram of the initial population generation process is shown in Figure 4. All virtual creatures in the population shared the same body morphology and neural network topology but differed in the values of the neural link weights. Thus, the genetic algorithm evolved the neural network weight values — a form of ANN training.

We defined vision infusion as a process of providing virtual creatures with simple sight. Through this process, virtual creatures gained vision sensors and neural networks that allowed them to gather information about source objects in the environment. The vision infusion process embedded sensory neurons in virtual creature body parts and linked them to the existing neural network as shown in Figure 5.

Up to three neurons were added to the main body part of each virtual creature: α-neuron, s_α signed d-neuron, and β-neuron. These neurons provided the ANN with the α angle measure, a combination of the signed s_α measure and the distance measure d, and the β angle measure, respectively. This neuron bundle represented a vision sensor centred on the main body part and sharing its coordinate axes.

The sensor neurons were linked to other valid neurons in the same body part and to valid neurons in direct child body parts. The weights of the neural connections were randomized. Figure 6 shows the neural network of an evolved virtual creature. The morphology and light following behaviour of this creature is shown in Figure 7.

5. RESULTS

In this section, we present the results in evolutionary training of walking virtual creatures in the light-following task

---

[2] Open Dynamics Engine (ODE) can be found at `http://www.ode.org`.

[3] OGRE can be found at `http://www.ogre3d.org`.

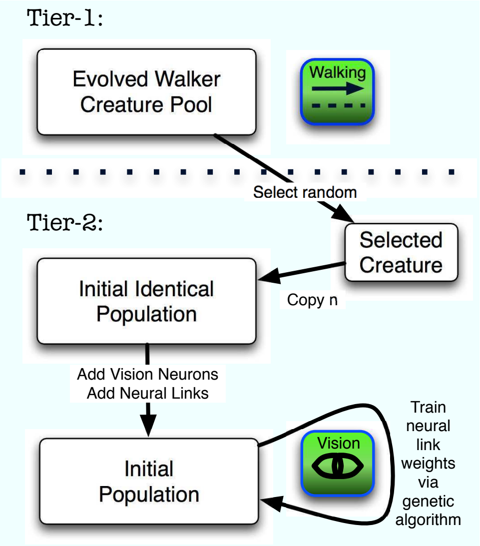


Figure 4: Schematic view of the vision cloning process used to generate initial populations in Tier-2 light following experiments. Previously evolved walking creatures were randomly selected, cloned, and infused with vision networks to fill an initial population for Tier-2 experiments.

using the methodology described in the previous sections. First, we take a look at the observed light following behaviours and creature morphologies from Tier-1 and Tier-2 experiments in light following. Following that, we present the results of experiments with different vision sensing neurons and discuss how the choice depends on the morphology and the seed creature behaviour.

5.1 Morphologies and Behaviours

Results of Tier-1 light following experiments (evolution of both the morphology and controller from random genotypes) were overall not satisfactory. While some creatures evolved light detection and light orientation behaviours, very few virtual creatures evolved light following. This poor performance might be due to the fitness function which does not incorporate the need for movement. Many of the virtual creatures did not move far from their original location.

In contrast to the Tier-1 results, most Tier-2 experiments (incremental evolution) evolved some form of light following behaviours ranging from simple turning toward the light to proper light following. Many of the behaviours observed were quite stable and not prone to sudden changes of creature body frame orientation (e.g., flipping on its side or back). Figure 7 shows the morphologies and light following behaviour of six evolved virtual creatures.

Several creatures were able to perform light following with varied body orientations. Some creatures performed well until their orientation changed due to their movement. They were then able to either fix their orientation and continue with proper light following behaviour or the fall rendered them unmovable. The reliance on the sensory information was clearly seen by disabling the light source. This triggered an evident change in the behaviours of the creatures, usually toward some kind of a random walk or fall back to their evolved Tier-1 behaviours.

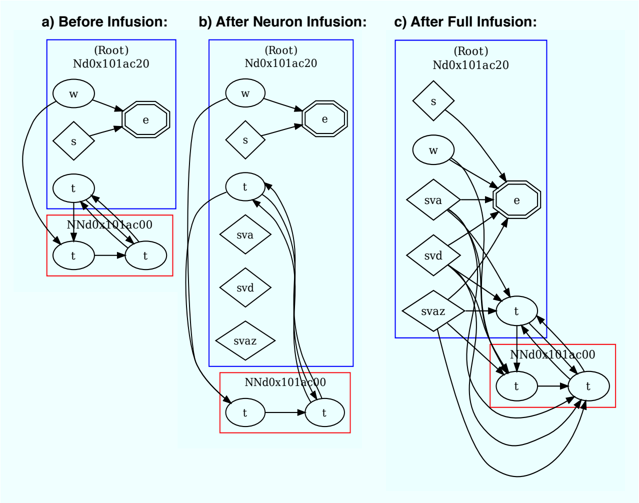


Figure 5: Illustration of the vision infusion process applied to a sample virtual creature neural network adding sensory neurons sva, svd, svaz and corresponding neural connections.

We analyzed the light following behaviours using two factors: light orientation and light strategy. Light orientation described the dynamics of the sensor orientation with respect to the light source. Creatures with *direct* light orientation were always trying to align their bodies toward the light (α value of 0). Those with *indirect* light orientation indirectly aligned toward the light (constant α value). The *varied* light orientation represented creatures that were able to perform light following with various body alignments.

The light strategy described the dynamics of the light following motion. Some creatures exhibited the *point-and-go* light strategy, an in-place turning behaviour to align themselves forming a direct or indirect light orientation before they were able to move toward the light. Other creatures used a *steer-toward* strategy which varied the light orientation while continuously moving toward the light. Finally, creatures with the *move-around* light strategy had complex, often seemingly random, movement toward the light.

Through observation, we categorized virtual creatures into three classes relating the evolved light following to the locomotion behaviour of the original seed creatures. Class 1 consisted of the seed creatures with stable movement orientations (usually swinging or pushing movement strategies) constrained to the simulation plane. These creatures quickly evolved light following behaviours even without the need for the β angle neurons (Figure 7(a–c)).

Creatures in Class 2 included those that evolved from seed creatures with a changing body orientation, usually coupled with a spinning movement strategy, but were able to adapt their neural network to movement with a stable orientation relative to the simulation plane. This allowed them to use the α angle and distance measures to effectively locate and navigate to the light source (Figure 7(d–e)).

Most of the evolved creatures in Class 2 used the steer-toward light strategy with either direct or indirect light orientation. Figure 7(e) shows an example of an adapted "Worm" creature which exhibited two stable movement ori-

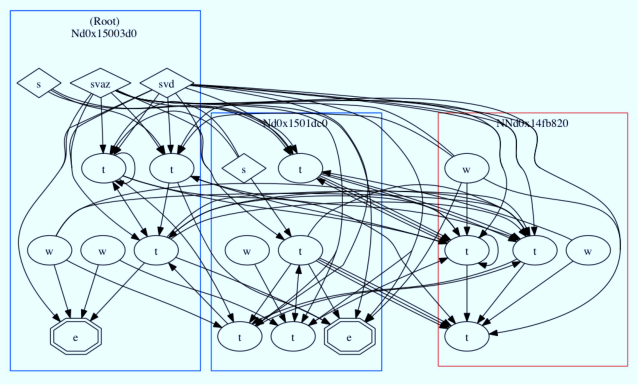


Figure 6: Neural network controller of the evolved virtual creature shown in Figure 7 (a). Neurons of various types (computational (t), periodic (w), sensor (s), effector (e), angle vision sensor (svaz), and distance vision sensor (svd)) are grouped in subnetworks embedded in two body nodes. The subnetwork on the right (red frame) represents the global neural node.

entations: with the sensor pointing up and with the sensor pointing parallel to the plane.

Finally, Class 3 consisted of creatures that had kept their changing orientation but still managed to evolve some light following behaviour. These creatures were often quite slow at reaching the light using a varied light orientation and the move-around light strategy as seen in Figure 7(f).

Since each experiment consisted of a whole population with morphologically identical creatures, the choice of the seed creature for each experiment limited its light following capabilities. Creatures with originally spinning or lifting movement strategies were often found in Class 3 in light following experiments. However, some of these, belonging to Class 2, were able to adapt their controllers for stable movement orientation and perform good light following behaviours. Finally, the creatures with stable body orientations and swinging or pushing strategies were found in Class 1 and were able to evolve very good light following behaviours.

5.2 Sensing Neurons

We performed experiments with combinations of the sensory neurons (α-neuron, s_α signed d-neuron, and β-neuron) which allowed for the analysis of the importance of each sensory neuron in the evolution of light following behaviour. Results of the experiments were compared based on the evolved light following behaviour and fitness performance.

We have observed better results with the s_α signed distance value compared to using both the α angle and distance. This might be due to a smaller neural network (i.e., no α angle neuron and corresponding links) which was easier to evolve using the genetic algorithm. We think that the sign of the α angle is enough for light detection and that the angle itself is an optimization feature (which, for instance, can be used to vary the body rotation speed with the angle).

The experiments without the β angle measure produced very few good light following behaviours. Proper light following occurred with creatures having a stable movement orientation or with those that were able to evolve orientation fixing behaviours. Thus, not enough information is provided to the neural network for creatures with changing orientations. Experiments with the addition of the β angle measure showed a large increase in proper evolved light following behaviours.

Figure 8 shows the fitness plots of four types of experiments with different vision sensor settings: (a) signed distance and α angle, (b) α and β angles, (c) signed distance and β angle, and (d) signed distance, α and β angles. The maximum population fitness (top curves, in red) and the average population fitness (bottom curves, in blue) are shown for 115 separate experimental runs. The envelope for each measure is shaded and the average over all runs is drawn with a thicker line.

The worst fitness performance was seen with the combined signed distance and α angle measure experiments. Figure 8(a) shows the low average fitness values and a constant trend in the average best-of-population fitness after 300 tournaments. Conversely, the β angle and distance measure combination produces the highest fitness and an increasing trend in average best-of-population fitness as seen in Figure 8(c).

The distance information is important to the neural network since the experimental runs without the distance measure did not perform well as shown in Figure 8(b). The distance measure provides the neural network with a slowly changing value that can promote more movement from the effectors thus creating more exploratory behaviours. Distance is also the main component of the fitness function.

The addition of the α angle degrades the performance of the experiments using the signed distance measure as illustrated in Figure 8 (c) vs. (d). The sign of the α angle that is incorporated into the distance measure seems to provide enough information to the neural network. The addition of the α angle measure adds complexity to the neural network with no clear benefit of the extra information.

6. CONCLUSION

We presented the results of experiments in evolution of virtual creature light following. In contrast to previous approaches (e.g., [12]), we used Morphid Academy to perform incremental evolution of light following on previously evolved virtual creatures. Through a process called Vision Infusion, we augmented the neural network controllers of selected virtual creatures and trained a population of seed creature clones using the light following task.

We demonstrated that it is feasible to use incremental evolution to evolve additional behaviours for previously evolved creature morphologies and neural network controllers. We were able to evolve several interesting light following behaviours, many of which were quite stable and self-repairing upon changes in creature orientation. We concluded that a simulation plane orientation angle, a relative location of the light source with respect to the creature orientation, and a distance measure provided enough information to evolve interesting light following behaviours.

The environment can play a crucial role in any evolutionary task. The light following experiments of Sims presented in [12] used either a land training environment (similar to our experiments) or an aquatic environment (full 3D). Although Sims does not differentiate between the environments while discussing the light following results, he does differentiate between virtual creatures that have learned the

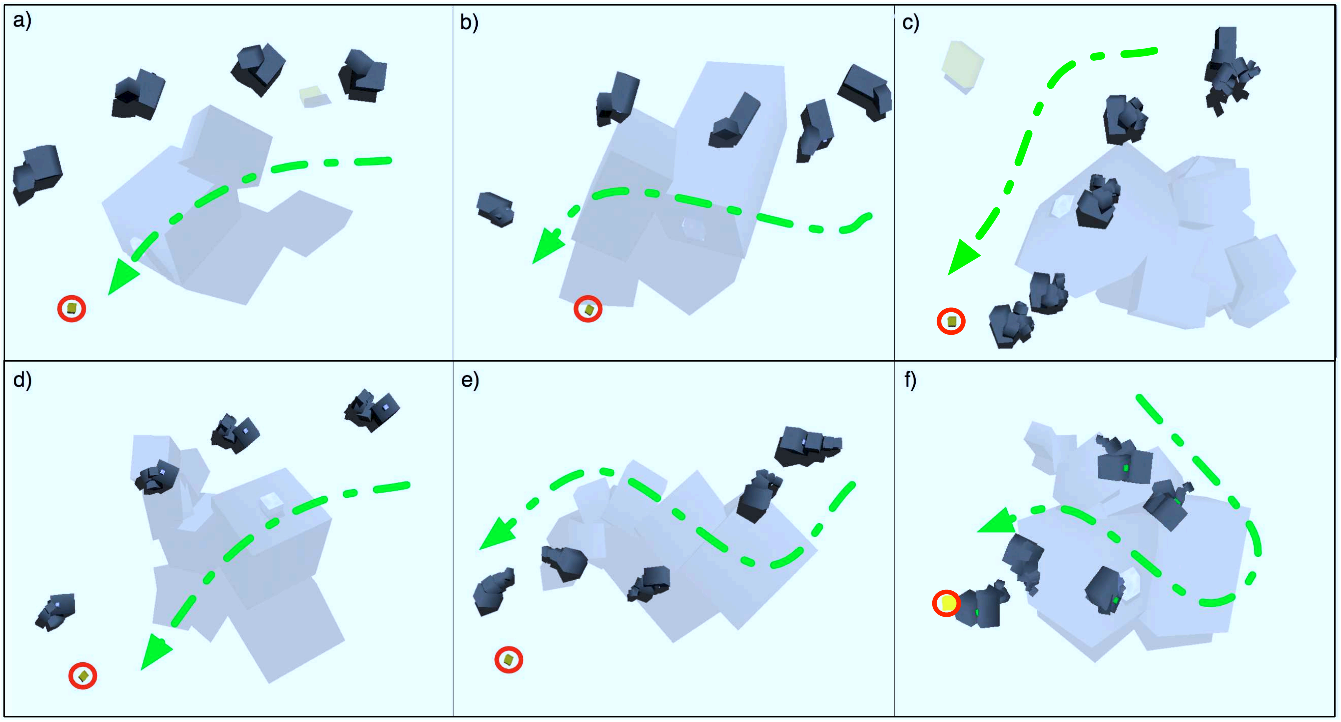


Figure 7: Sample evolved creature morphologies and the corresponding light-following behaviours. The creatures are categorized into three classes: 1) constrained to the plane (a–c), 2) adapted to the plane (d–e), and 3) random (f). Time lapse images are superimposed to show the path and orientation of each creature. The light source is circled in red. The background image shows a close-up of each creature.

behaviour well and others that have learned it partially or inconsistently. A full 3D environment should produce better light following behaviour due to unconstrained motion of the virtual creatures and provides an extension to this work.

Our fitness functions relied on a distance or displacement measure toward the light source. While such a measure works well on virtual creatures that have already learned to move around their environment, the lack of a benefit for movement can retard the learning process for initially random virtual creatures. We think that incorporating a movement component (e.g., through a distance measure) to the fitness function of Tier-1 experiments will create better virtual creatures that are able to perform proper light following behaviours.

The evolutionary algorithm used in our experiments only performed mutation of the virtual creature neural network link weights. We performed initial experiments with structure preserving crossover with inconclusive evidence of better performance. Further work is required in genetic operators that modify the structure of the neural network. The addition of a learning component that is able to train the ANN while it is evolving is also of interest.

Similar experiments to the ones we presented can evolve the controllers for physical mobile robots. Robotic morphology designs can also be evolved, especially for locomotion based tasks. Neural network training through evolutionary algorithms is important for mobile robotics in situations where the design of the robot is predetermined or in environments that can unexpectedly change the orientation of the robot or even change the morphologies themselves.

The results presented in this paper form the ground work for future experimentation in the study of functional form evolution. The resulting virtual creatures can be further simulated in co-operative and ecosystem scenarios where the vision functionality is necessary to avoid environmental obstacles, to find resources, and to find or avoid other creatures. Such experiments can provide insight into the complex processes of the evolution of functional body forms and can allow for the emergence of open-ended evolution.

The open-source Morphid Academy system is cross platform and written in C++. All the experiments in this study were performed on a cluster of 50 Apple Mac minis, each with a 1.66GHz Intel Core Duo processor and 1GB of RAM. Further information about the Morphid Academy and the experimental results can be found at the Morphids Project website at `http://www.morphids.com`.

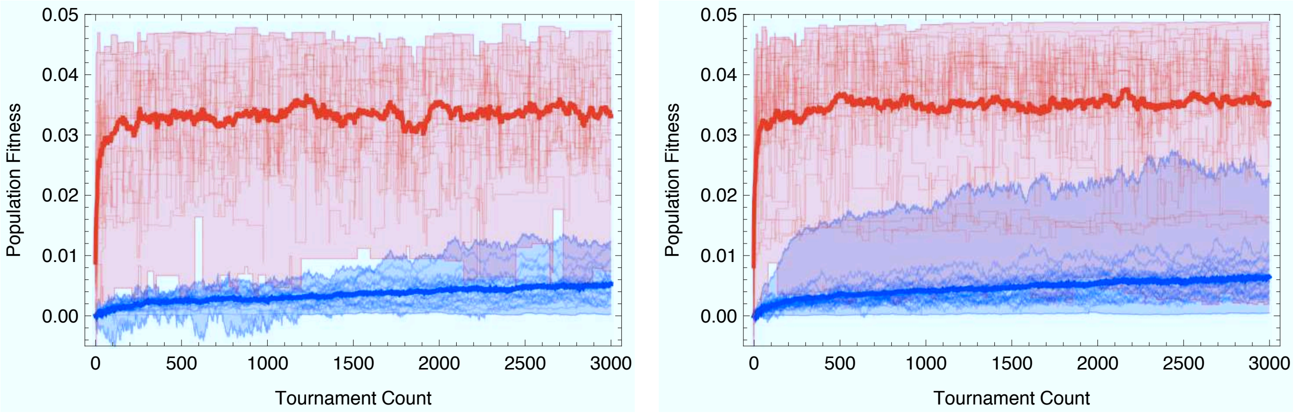
7. REFERENCES

[1] J. Bongard. Behavior Chaining: Incremental Behavior Integration for Evolutionary Robotics. In *Artificial Life XI*, Cambridge, MA, 2008. MIT Press.

[2] R. A. Brooks. Intelligence without Representation. *Artificial Intelligence*, 47:139–159, January 1991.

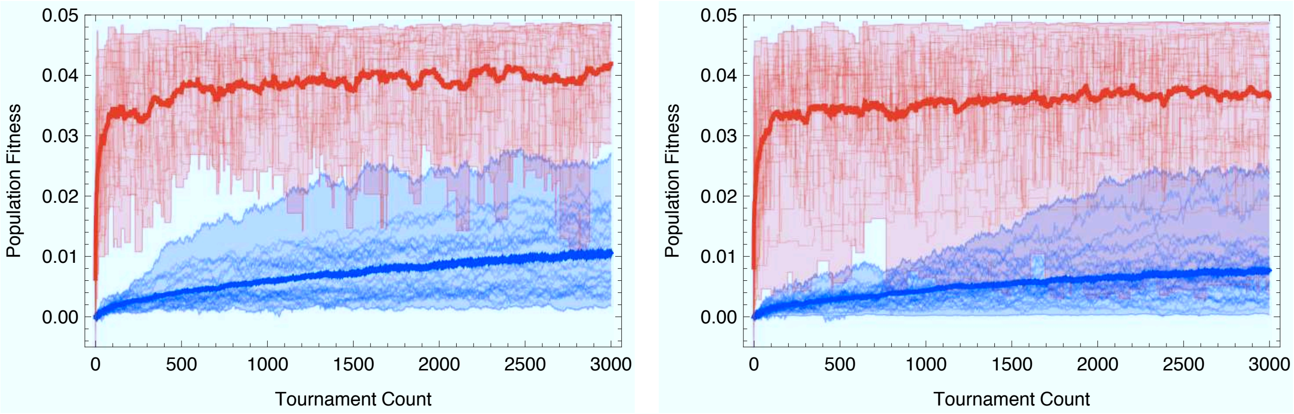
[3] N. Chaumont, R. Egli, and C. Adami. Evolving Virtual Creatures and Catapults. *Artificial Life*, 13:139–157, 2007.

[4] A. L. Christensen and M. Dorigo. Incremental Evolution of Robot Controllers for a Highly Integrated Task. In *Proceedings of The Ninth International Conference on the Simulation of Adaptive Behavior*



(a) signed distance + α angle

(b) α angle + β angle



(c) signed distance + β angle

(d) signed distance + α angle + β angle

Figure 8: Fitness plots for light-following experiments with sensory neuron combinations (a) through (d). Population curves from 115 independent experiments are plotted against the tournament count. The top curves (colored in red) represent the maximum population fitness and the bottom curves (colored in blue) represent the average population fitness. The envelope for each measure is shaded and the average over all runs is drawn with a thicker line.

SAB'06, pages 473–484, Berlin, Germany, 2006. Springer-Verlag.

[5] M. Dorigo. SWARM-BOT: An experiment in swarm robotics. In P. Arabshahi and A. Martinoli, editors, *Proceedings of SIS 2005 – 2005 IEEE Swarm Intelligence Symposium*, pages 192–200, Piscataway, NJ, 2005. IEEE Press.

[6] T. Miconi. Evosphere: Evolutionary Dynamics in a Population of Fighting Virtual Creatures. In *CEC2008: Proceedings of the IEEE Congress on Evolutionary Computation*, pages 3066–3073, 2008.

[7] T. Miconi and A. Channon. An Improved System for Artificial Creatures Evolution. In e. a. L.M. Rocha, editor, *ALIFE X: Proceedings of the 10th Conference on the Simulation and Synthesis of Living Systems.* MIT Press, 2006.

[8] S. Nolfi and D. Parisi. Evolution of Artificial Neural Networks. In *In Handbook of brain theory and neural networks*, pages 418–421. MIT Press, 2002.

[9] M. L. Pilat. *Morphid Academy: A Virtual Laboratory for Evolution of Form and Function.* PhD thesis, University of Calgary, 2009.

[10] M. L. Pilat and C. Jacob. Creature Academy: A System for Virtual Creature Evolution. In *CEC2008: Proceedings of the IEEE Congress on Evolutionary Computation*, pages 3289–3297. IEEE Press, 2008.

[11] K. Sims. Evolving 3D Morphology and Behavior by Competition. *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, 1(4):353–372, 1994.

[12] K. Sims. Evolving Virtual Creatures. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22, New York, NY, USA, 1994. ACM Press.

[13] M. H. Wake. Bodies and Body Plans, and How They Came to Be. In W. J. Kress and G. W. Barrett, editors, *A New Century of Biology*, pages 29–52. Smithsonian Books, 2001.

[14] J. Walker, S. Garrett, and M. Wilson. Evolving controllers for real robots: A survey of the literature. *Adaptive Behavior*, 11(3):179–203, 2003.