ARTIFICIAL LIFE TECHNIQUES FOR GENERATING CONTROLLERS FOR PHYSICALLY MODELLED CHARACTERS

Tim Taylor
International Centre for Computer Games and Virtual Entertainment (IC CAVE)
University of Abertay Dundee
Bell Street, Dundee DD1 1HG
Scotland
E-mail: tim.taylor@abertay.ac.uk

KEYWORDS

AI, Artificial Life, Physics and Simulation

ABSTRACT

The realistic physical modelling of characters in games and virtual worlds is becoming a viable alternative to more traditional animation techniques. Physical modelling can enhance realism and allow users to interact with the world much more freely. However, designing controllers to move physically modelled characters (e.g. to make a human character walk) is generally a difficult task. Artificial life techniques can be useful in automating this task. For example, artificial evolution can generate suitable controllers for simple behaviours, given only a high level description of that behaviour in terms of a fitness function. In this paper, the state of the art in evolving controllers, and also in evolving the characters' body shapes, is described. It is then suggested that current approaches will not be able to scale up to more complicated behaviours. A number of possible solutions to this problem are described. Finally, several routes for delivering this technology to the games industry are discussed.

INTRODUCTION

There is a move in the computer games and digital entertainment industries away from traditional animation techniques and towards accurate physical modelling of characters and environments. By modelling the forces and torques acting on bodies within a virtual environment, detecting and responding to collisions between bodies, respecting constraints arising from, for example, joints in articulated multi-body objects, etc., the system behaves in a believable manner in all situations, and the user of the system can therefore be allowed to interact with it much more freely. The growing popularity of this approach is demonstrated by the appearance over the last couple of years of a number of off-the-shelf physics engines aimed at games programmers (e.g. MathEngine[1], Havok[2], Ipion[3]). While these engines handle the modelling of inanimate bodies, programmers are still left with the task of writing controllers for motile objects within the environment (e.g. cars, human characters, monsters, etc.). Writing a controller for a physically modelled character is a question of calculating the appropriate forces to apply to each body part at each point of time to achieve the desired movement (e.g. a realistic walking gait for a human character). This is generally a very hard task.

There is therefore some interest in the idea of automating the generation of controllers for physically modelled characters. In this paper I will present an overview of the state of the art in evolving controllers and body shapes for physically modelled characters, and describe the design of such a system. I claim that current techniques are, however, limited in terms of the complexity of behaviours they may be expected to produce. I suggest a number of ways in which such systems may be improved to generate more complex creatures and behaviours. I conclude by suggesting a number of ways in which this sort of technology may best be utilised by the games industry.

STATE OF THE ART

A number of groups have employed techniques from the fields of artificial intelligence and

[1] http://www.mathengine.com
[2] http://www.havok.com
[3] http://www.ipion.com

artificial life to automate the generation of controllers for physically simulated characters. In this paper I will focus on work that employs genetic algorithms or similar artificial evolutionary methods to generate controllers, although the use of other adaptive techniques may certainly also be fruitful, e.g. (Terzopoulos et al. 1994).

The use of genetic algorithms in this area generally falls into one of two categories: the evolution of controllers for characters with predefined, fixed body shape; and the evolution of both the controller and the body shape of the creature concurrently. Examples of the former category include (van de Panne and Fiume 1993; Gritz and Hahn 1997; Arnold 1997; Reeve 1999; Reil 1999; Ijspeert 2000). Examples of the latter category of evolving both controllers and body shapes include (Sims1994b; Ventrella 1999; Komosinski and Ulatowski 1999; Taylor and Massey 2000; Ray 2000; Lipson and Pollack 2000; Bongard and Paul 2000).

In the remainder of this section I will briefly describe the design and implementation of a system for evolving both the controller and body shape of a character. The evolution of controllers for characters of fixed shape can be achieved in a similar manner, although in this case the genetic descriptions should obviously only describe the characters' controllers, and not their shapes. Full design details can be found in the references given for specific works.

The seminal example of such a system was that developed by Karl Sims (Sims 1994b). He used an artificial evolutionary process to generate both controllers and body shapes for creatures that could perform simple tasks such as crawling, swimming, jumping and following a target. Sims' original work, published in 1994, employed a Connection Machine CM-5 parallel computer, on which he developed his own code for performing the physical modelling of the creatures. However, desktop PCs are now powerful enough to run these sorts of systems in reasonable time.

Each creature is described by a "genome" that contains information about the body shape and controller. In Sims' work, the body shape was described by a directed graph, where each node represents an individual body part, and the connections between nodes describe how the parts are connected. Each node also describes an augmented neural network type controller for the corresponding body part. This representation provides modularity to the mapping from genotype (the description of the creature) to phenotype (the instantiation of the creature as a physical model and controller), and naturally leads to features such as duplication and recursion of body parts.

A run is started by randomly generating a population of genotypes. Each genotype in turn is translated into a physical creature, and then evaluated in a physically simulated environment for its performance at a given task. Sims used two basic environments, sea and land. The sea environment included a simplistic model of fluid drag, while the land environment included gravity, a ground plane, and frictional forces for ground contacts. A number of different criteria (or "fitness functions" in genetic algorithm jargon) were used for scoring the success of each creature in its environment, but they all basically rewarded creatures for movement.

The first, randomly generated population of creatures typically performed poorly at the designated task, although a few would, by chance, have some degree of success. Each creature was scored according to its performance, and when all creatures had been evaluated the population was ranked according to score. The best individuals were kept to form the basis of a new generation. This new population was filled up by adding mutated forms of these best genotypes and genetic crosses of pairs of genotypes (i.e. new genotypes were formed from the combination of parts from two different parent genotypes).

By repeating this process, efficient creatures evolved surprisingly quickly. Sims typically used populations of 300 individuals and ran the experiments for 50-100 generations. In a recent reimplementation of Sims' work, a single run of this size would take between 4-8 hours on a single 400MHz Celeron PC (Taylor and Massey 2000). Pictures and movies of a variety of creatures evolved using these systems are available online[4].

One of the nice properties of systems such as these is that each run generally produces a very different result due to the stochastic nature of the evolutionary process. Sims only selected

[4] Pictures of Sims' evolved creatures can be seen at http://www.genarts.com/karl/evolved-virtual-creatures.html; pictures and movies of Taylor and Massey's reimplementation are at http://computing.tay.ac.uk/timtaylor/demos/mathengine/

for high-level behaviours such as the ability to move forwards, but within the vast space of different creature designs describable with the genetic system used, there are countless forms which can competently perform such behaviours. The evolutionary process is therefore a tool for exploring interesting regions of this immense landscape of creature designs; it is a creative machine for generating suitable and interesting forms and behaviours, not limited by the preconceptions of a human designer' s imagination.

LIMITATIONS OF CURRENT SYSTEMS

The work referred to in the previous section demonstrates that artificial evolution is a promising technique for automating the process of controller (and body shape) generation. However, the complexity of behaviours that have been generated so far has been somewhat limited; most of the work has produced characters that can walk, crawl, swim, jump, or, at best, follow a moving target. These results beg the question of whether the approach will scale to the generation of more complex behaviours.

The major problem with using a standard genetic algorithm, as described in the previous section, is that the goal behaviour to be (hopefully) evolved must be expressed in the form of a fitness function that is used to score each creature. The evolutionary process can only get off the ground if the initial, randomly generated population of creatures contains at least one creature (and preferably more) that can achieve a non-zero score on the fitness function. The more complicated the function is, the less likely this is to be the case. For example, consider an attempt to evolve a behaviour whereby a creature needed to process complex visual data about the movement of another creature standing in front of it, and use this to decide whether that creature was a friend or foe. This enterprise would clearly have little chance of success if the evolutionary process was starting from scratch.

Researchers in the field of evolutionary robotics have considered various methods of overcoming these difficulties. These generally involve either the incremental acquisition of increasingly complicated tasks, e.g. (Dorigo and Colombetti 1997; Perkins 1998), or the decomposition of the task into a sequence of easier tasks together with some way of combining them, e.g. (Lee 1997). A problem with many of these approaches is that the decomposition of the task into easier and/or incremental steps is something of an art in itself; there are no general guidelines to suggest the most appropriate way to do this. What may seem like a sensible task decomposition from the designer's point of view may not be the best route by which to *evolve* a complex behaviour; in the worst case, it could even be a hindrance.

Work in this area is on-going, but it has yet to be proved whether these approaches can usefully be employed to evolve complex behaviours in the general case.

ACHIEVING MORE COMPLEX BEHAVIOURS

Despite the problems described above with evolving single creatures to perform complex tasks, there are a number of alternative approaches that have shown some signs of success. In this section I will describe five of these.

Co-Evolution

The genetic algorithms community has long known the potential of using *co-evolution* to improve the quality and complexity of evolved solutions. The idea is that rather than evolving a population against a fixed fitness function, two populations are used instead, with one evolving against the other. For example, Hillis evolved efficient number sorting algorithms by co-evolving a population of candidate algorithms against a population of numbers to be sorted (Hillis 1991). As the sorting algorithms improved, so the population of numbers to be sorted evolved to present tougher challenges to the algorithms. Evolution in one population thereby catalysed further evolution in the other.

Sims used the same technique to evolve interesting competitive behaviours in virtual creatures (Sims 1994a). In this work, rather than simulate a single creature at a time, he simulated a couple. These two creatures were opponents in a simple game where they had to fight for possession of a cube that was initially placed halfway between them. The creatures not only evolved behaviours to reach the cube quickly, but also to fend off their opponents. A wide variety of different behaviours were evolved in this way, and the resulting competitions were very engaging to watch.

Despite Sims' results, I am unaware of any subsequent work, by Sims or anyone else, in co-evolving both controllers and body shapes of competing creatures[5]. Co-evolution is certainly a very promising technique for developing complex behaviours, especially where these behaviours take the form of a competition between two or more creatures.

Virtual Ecologies

Taking co-evolution to its logical conclusion, we could forget about defining some sort of high-level game (such as fighting for possession of a cube) altogether. Instead, we could simply concurrently simulate a large population of creatures that are competing with each other for basic resources such as food and space, and let the creatures reproduce "by themselves" according to, say, how much food they have managed to eat. That way we get away from the sort of "artificial selection" we have talked about up to now (a fitness function is really the virtual equivalent of the selection employed by livestock farmers when trying to improve a particular trait in their animals), towards a closer approximation to "natural selection". We would not be selecting for particular high-level behaviours, but rather creating a virtual ecology and letting it evolve in any way it likes.

Some progress in this direction has been made, e.g. (Ventrella 1999; Komosinski and Ulatowski 1999), although because of the much larger drain on computational resources required to simulate large populations of creatures concurrently, these studies have used somewhat simplified physical simulations. As computers become increasingly powerful, as physics engines become more efficient, and, most importantly, with the opportunities for running massively parallel simulations afforded by the Internet, the size of the ecologies that can be modelled and the complexity of the physical simulation will certainly increase greatly over the coming months and years.

Lifetime Learning

In the systems described in this paper, the creatures' controllers improve by evolution, but they do not actually adapt whilst an individual creature is being simulated (i.e. over a creature's "lifetime").

Some recent results from evolutionary robotics suggest that combining an evolutionary algorithm with the ability of the controllers to adapt or learn over an individual creature's lifetime can lead to improved robustness and complexity of behaviours compared to evolution by itself, e.g. (Floreano and Urzelai 1999; Kondo et al. 1999).

Giving an individual creature the ability to adapt and learn during its lifetime effectively smoothes the search space over which evolution is happening, thereby helping the process to progress (Maynard Smith 1987). It is reasonable to assume that adding these sorts of abilities to our artificial creatures will improve their ability to evolve complex behaviours just as it has done in evolutionary robotics.

Behavioural Primitives

In the previous section, various methods were described for achieving the evolution of complex behaviours by task decomposition. While it has yet to be demonstrated that complex behaviours can, in general, be *evolved* in this manner, related approaches could be more successful.

For example, rather than trying to evolve complex behaviours, another approach is to evolve a collection of primitive behaviours, and then use other, non-evolutionary techniques for combining these primitive into more complicated sequences.

The task of programming a game character using this approach could now become somewhat like directing an actor in a film; the character has a certain degree of competence at autonomous behaviour (some actors have more than others!), and the programmer/director issues high-level commands to it such as "run to the door!" or "attack the human! ". Some have suggested that this is the most useful route by which this sort of technology may be harnessed by games programmers, e.g. (Stern 1999). Alternatively, rather than directing the creatures manually, other more traditional AI techniques could be used to co-ordinate the primitive behaviours to automatically generate more complex tasks.

Various examples exist of the potential of this kind of approach, although these have used methods other than evolution to produce the primitive behaviours, e.g. (Terzopoulos et al. 1994).

[5] Although some work has been done on co-evolving controllers for robots of fixed morphology in very simple environments, e.g. (Cliff and Miller 1996).

User Guided Evolution

Another alternative to supplying a fixed fitness function to the genetic algorithm is to present the user with a variety of creatures from the evolving population at various intervals, and allow them to select their favourite creatures to be used as the foundation of the next generation. The user may select the creatures by any criteria they wish, and can therefore guide the path of evolution according to their own preferences without having to formally express these preferences. Ray has recently used this technique to evolve virtual pets according to users' aesthetic choices (Ray 2000). This technique allows users to participate in the evolutionary process and therefore feel more attached to, and more empathy towards, the evolved creatures.

USING THE TECHNOLOGY

The computational expense of evolving physically modelled creatures is undoubtedly a major reason why this technology has not yet been adopted by the computer games industry at large. The problems with evolving complex behaviours, as described in this paper, are also a factor. The former problem is already starting to disappear with the continued increase in available computing power, and it is more than likely that the latter problem can be overcome, for example in the ways outlined in the previous section.

However, even ignoring these problems, the question of how this technology might best be delivered to the games industry remains. At one end of the spectrum, it could be that it is most productive for specialist consultants with experience of the technology to evolve bespoke creatures for games companies on request, according to the company's specifications. At the other end of the spectrum, an application could be delivered that allows end-users to evolve their own creatures. A number of alternative strategies can be imagined in between these two extremes. In this section I will briefly describe three possible routes, although this is by no means an exhaustive list.

Licensing Characters from a Virtual Zoo

One possibility would be for a specialist artificial life company to develop this technology in-house, and build up a huge library, or virtual zoo, of different characters. Games companies could inspect these characters and licence individual characters for specific games or other applications. This approach would be particularly appropriate for evolved body shapes as well as controllers, as characters evolved in this way often have very strange or unusual forms and behaviours. The virtual zoo could therefore act as a source of inspiration for new types of characters, as well as a source for directly licensable characters. There would undoubtedly be some intellectual property issues to be overcome if this approach is to be commercially viable.

Online Evolution

At the other extreme, a consumer application could be written that allowed users to evolve their own creatures, and to exchange them with other users over the Internet. Alternatively, a central server could oversee a genetic algorithm in which the evaluation of individual creatures was farmed out to end-users' PCs over the Internet. This could be achieved as a background process on the PC, while the user was playing a game, or it could be run as a screen-saver. Users could also be allowed to guide evolution according to their own aesthetic tastes, as described in the previous section. A whole range of other possibilities is imaginable. The trick would be to develop an application with an understandable user interface to the evolutionary process, and one that inspired the user's interest and curiosity.

Co-Evolved Characters for Combat Games

A final example application would be to co-evolve twin populations of creatures competing in some sort of contest, such as a combat game. Having co-evolved creatures in this way, by allowing evolution in one population to catalyse evolution in the other (as described in the previous section), one of the creatures in the contest could simply be replaced by a human player. The difficulty here would be to design a user interface that allowed the player to control the creature in a suitable way.

CONCLUSION

Physical modelling is becoming an increasingly important technology in the games industry. With it comes the need for automated ways of generating controllers for physically modelled characters. A number of studies have demonstrated the potential of artificial evolution for this purpose. Using physical modelling and artificial evolution also allows us to evolve the body shapes of the characters, not just their controllers. It is therefore a very powerful technique for exploring potential behaviours in a given environment. However, the studies to date have only dealt with the generation of fairly simple behaviours, and it is questionable whether a straightforward genetic algorithm by itself would be able to evolve more complex behaviours.

In this paper, a number of potential methods for overcoming this problem have been suggested. These include methods to enhance the dynamics of the evolutionary process (i.e. co-evolution, virtual ecologies, lifetime learning), a method for integrating this technology with other techniques (using behavioural primitives), and a method for allowing users to guide the course of evolution according to their own tastes.

A number of possible ways of delivering this technology have also been described, although these all have a number of obstacles to be overcome before they are truly viable. However, with more games companies using physical modelling (either with their own code or with one of the available off-the-shelf physics engines), and with more researchers looking at artificial life techniques for generating controllers, we can expect to see this technology start to appear in commercial products in the very near future.

REFERENCES

Arnold, D. (1997). *Evolution of Legged Locomotion.* MSc thesis, School of Computing Science, Simon Fraser University.

Bongard, J.C. & Paul, C. (2000). Investigating Morphological Symmetry and Locomotive Efficiency Using Virtual Embodied Evolution. Meyer, J.-A. et al. (eds.) *From Animals to Animats: The Sixth International Conference on the Simulation of Adaptive Behavior*: MIT Press.

Cliff, D. & Miller, G. F. (1996). Co-Evolution of Pursuit and Evasion II: Simulation Methods and Results. Maes, P. et al. (eds.) *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96)*, pp.506-515: MIT Press.

Dorigo, M. & Colombetti, M. (1997). *Robot Shaping: An Experiment in Behavior Engineering.* MIT Press.

Floreano, D. & Urzelai, J. (1999). Evolution of Neural Controllers with Adaptive Synapses and Compact Genetic Encoding. Floreano, D. et al. (eds.) *Advances in Artificial Life: Proceedings of the Fifth European Conference on Artificial Life*, pp.183-194: Springer Verlag.

Gritz, L. & Hahn, J. K. (1997). Genetic Programming Evolution of Controllers for 3-D Character Animation. Koza, J.R. et al. (eds.) *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pp.139-146: Morgan Kaufmann.

Hillis, W. D. (1991). Co-evolving parasites improves simulated evolution as an optimization technique. Langton, C. G. et al. (eds.), *Artificial Life II*, pp.313-384: Addison-Wesley.

Ijspeert, A.J. (2000). A 3-D Biomechanical Model of the Salamander. Heudin, J.-C. (ed.) *Proceedings of the Second International Conference on Virtual Worlds*, pp.225-234: Springer Verlag.

Komosinski, M. & Ulatowski, S. (1999). Framsticks: Towards a Simulation of a Nature-Like World, Creatures and Evolution. Floreano, D. et al. (eds.) *Advances in Artificial Life: Proceedings of the Fifth European Conference on Artificial Life*, pp.261-265: Springer Verlag.

Kondo, T., Ishiguro, A., Tokura, S., Uchikawa, Y. & Eggenberger, P. (1999). Realization of Robust Controllers in Evolutionary Robotics: A Dynamically-Rearranging Neural Network Approach. *Proceedings of 1999 Congress on Evolutionary Computation*: IEEE.

Lee, W.-P. (1997). *Evolving Robots: from Simple Behaviours to Complete Systems.* Ph.D. thesis. Department of Artificial Intelligence. University of Edinburgh.

Lipson, H. & Pollack, J.B. (2000). Automatic Design and Manufacture of Robotic Lifeforms. *Nature* 406, pp.974-978.

Maynard-Smith, J. (1987). When Learning Guides Evolution. *Nature* 319, pp.761-762.

Perkins, S. (1998). *Incremental Acquisition of Complex Visual Behaviour using Genetic Programming and Shaping*. PhD thesis, Division of Informatics, University of Edinburgh.

Ray, T.S. (2000). Aesthetically Evolved Virtual Pets. Maley, C.C. and Boudreau, E. (eds.) *Artificial Life 7 Workshop Proceedings*. 158-161. Online version: http://www.hip.atr.co.jp/~ray/pubs/alife7a/

Reeve, R. (1999). *Generating Walking Behaviours in Legged Robots*. PhD thesis, Division of Informatics, University of Edinburgh.

Reil, T. (1999). *Artificial Evolution of Neural Controllers in a Real-Time Physics Environment*. MSc thesis, COGS, University of Sussex.

Sims, K. (1994a). Evolving 3D Morphology and Behavior by Competition. Brooks, R. and Maes, P. *Artificial Life IV Proceedings*, pp.28-39: MIT Press.

Sims, K. (1994b). Evolving Virtual Creatures. Computer Graphics (*SIGGRAPH 94 Proceedings*), pp.15-22: ACM SIGGRAPH.

Stern, A. (1999). AI Beyond Computer Games. Dobson, D. W. & Forbus, K. (eds.) *Artificial Intelligence and Computer Games: Papers from 1999 Spring Symposium*: AAAI Press.

Taylor, T. & Massey, C. (2000). *MathEngine Evolved Creatures.* Online demo available at http://computing.tay.ac.uk/timtaylor/demos/mathengine/

Terzopoulos, D., Tu, X. & Grzeszczuk, R. (1994). Artificial Fishes: Automomous Locomotion, Perception, Behavior, and Learning in a Simulated Physical World. *Artificial Life* 1, pp.327-351.

Van de Panne, M. & Fiume, E. (1993). Sensor-Actuator Networks. Computer Graphics (*SIGGRAPH 93 Proceedings*), pp.335-342: ACM SIGGRAPH.

Ventrella, J. (1999). Animated Artificial Life. Chapter 3 in Heudin, J.-C. (ed.) *Virtual Worlds: Synthetic Universes, Digital Life and Complexity*: Perseus Books.

AUTHOR BIOGRAPHY

Tim Taylor studied Natural Sciences at Cambridge University, where he specialised in Experimental Psychology, focusing on neurophysiological and cognitive mechanisms for intelligent behaviour. He gained a broad understanding of living systems by taking courses in the Biology of Cells, Geology/Paleobiology, and the History and Philosophy of Science, where he concentrated on the life and work of Charles Darwin. He also took courses in Physics and Mathematics.

After graduating from Cambridge in 1992, he moved to Edinburgh University to take an MSc in Artificial Intelligence, which he passed with distinction. After a couple of years working as a professional C++ programmer in London, he returned to Edinburgh in late 1995 to start working for a PhD. His thesis, ‘From Artificial Evolution to Artificial Life”, was an empirical investigation of the dynamics of a system of self-replicating, evolving computer programs, and in it he discussed reasons why the evolutionary potential of such systems appeared to be somewhat limited, plus ways in which these limitations may be overcome. Tim was awarded his PhD in 1999.

In June 1999 he started working for MathEngine, developing artificial life techniques for generating controllers for physically modelled characters. Wishing to return to academia, he moved to the University of Abertay Dundee in March 2000, to take up a position as a postdoctoral research associate in the newly formed International Centre for Computer Games and Virtual Entertainment. He is currently looking at increasing the behavioural complexity of physically modelled characters that can be generated using artificial life techniques, and also at a variety of ways of delivering this technology to the computer games and entertainment industries.