Using a General Purpose Virtual Environment for Artificial Life Simulations

Cesar Gomes Miguel, Marcio Lobo Netto
Department of Electronic Engineering
Cognitive Sciences Research Group
University of São Paulo
{fcesargm}, {lobonettg}@lsi.usp.br

Abstract
This short paper describes a general purpose opensource 3D virtual environment called BREVE. We also briefly describe a recent method (NEAT) for the evolution of arbitrary topology artificial neural networks. This method is used in conjunction with BREVE in a simple Artificial Life simulation where a population of virtual organisms controlled by neural networks evolves to accomplish a given task. The role of a good and freely available virtual environment package for such experiments is emphasized using a simple philosophy design: there is no need to reinvent the wheel.

1. Introduction
Artificial Life (ALife) is a multidisciplinary field concerned with building biological models that synthesizes some basic aspects of life [1]. Those models are basically defined by the interaction of many independent sub parts, often constituted of very simple rules that, when observed globally, shows properties of emergent behavior. For instance, a single ant has a very limited number of behaviors but a population of ants, an ant hill, acts like a single living organism.

The approach of ALife makes a heavy use of computer simulations and often demands the use of a tool for visualizing the resulting emergent behavior. Moreover, the simulated virtual environment needs to reproduce to some degree of precision the main properties of the real world, like the basic laws of physics and optics. In short, besides modeling each agent's behavior with local rules (how to behave under certain conditions, like the ants), we need a virtual world to place the interacting agents and observe the results, or as in our previous example, the ant hill. Most of these models work as an evolvable multi-agent system, in which the agent is an artificial organism living in a virtual environment.

In this work, the local rules of behavior is evolved using NEAT [6] and the simulated virtual environment will be handled by BREVE [2].

1.1 A View Of The Problem
In several works related to ALife the virtual environment is developed and maintained by the authors or by a small group of users, often restricting its usage for the community of researchers interested in the same subject. In addition, it is worth noting that developing a system for the task requires a great amount of work and knowledge in diverse fields such as computer graphics and advanced programming. In cases where the code is released to the public we are often faced with the loss of good and properly documented work.

This short paper is divided in three small sections. First we describe a neuroevolution method, known as NEAT [6]. This method applies a modified genetic algorithm to evolve arbitrary recurrent neural networks using a special type of chromosome encoding. Secondly, we give a short description of BREVE, an advanced opensource software package for general purpose 3D visualization and simulation of multi-agent systems [2]. For the last section, a NEAT Python implementation is used along with BREVE showing how an artificial life simulation where a population of simple neural network-controlled organisms evolve to solve a given problem.

1.2 Related Work
As stated in section 1.1, there have been several artificial life experiments where the virtual environment was specifically designed for the problem. Probably the most well known work is of Karls Sims [5], a virtual world in which artificial creatures evolve to solve a certain type of task, like grab a cube or swimming.

A more recent work investigated the use of a genetic algorithm to evolve the behavior of a virtual robot, controlled by a finite state machine [3]. The environment consisted of an arena enclosed by walls and a robot had to evolve to classify some randomly positioned objects.
Another attempt was a framework for ALife experiments called ALIVE [4], developed in Java 3D and aimed to be used as a virtual laboratory in which different types of virtual environment scenarios could be simulated.

2. Evolving Artificial Neural Networks

Most simulations embracing the use of neural networks as controllers are restricted to fixed topologies and the only characteristic that distinguishes them in a population is their synaptic weights. This is usually represented by a vector of float numbers and can be easily used as a chromosome encoding that single organism. However, the topology restriction implies that the number of possible behaviors is limited to a subset that may not contain the desired one. It is also unrealistic from the biological point of view since the nervous systems in nature have evolved through millions of years, causing a higher adaptation of certain species and eventually reaching human intelligence. A recent model known as NEAT [6] (NeuroEvolution of Augmented Topologies) was able to surpass this limitation using a sophisticated genetic algorithm that works with chromosomes of variable length and allows us to evolve real complex neural networks with arbitrary topology.

Basically, the model uses a special and general neural network encoding technique (Figure 1) in which it is possible to apply mutation and crossover genetic operators without destroying the parent’s offspring (a problem known as competing conventions [6]). The process of adding new genes (that can be a link or node gene) through mutations (and thus augmenting the neural network) is called complexification (Figure 2). It is very likely that with the addition of new structures the offspring will have an inferior fitness value relative to its parents and would probably be taken out the population by selection. A speciation method is applied to ensure that new innovations can be retained in a separated species and given enough time to adapt before competing with others for resources. If after a specified number of generations the species does not improve its average fitness, all its members are replaced by the new offspring.

It should also be noted that since a neural network is a particular kind of graph structure (where neurons are the nodes and synapses are the edges), anything ranging from a Bayesian Network to a Finite State Machine can be evolved with NEAT.

Although NEAT has several open-source implementations in different programming languages available on the internet, in this work we use a recent Python implementation that simplifies the work for writing complex simulations¹.

3. The Breve Environment

Specially designed for multi-agent systems simulations, BREVE2 is a rich and open-source 3D virtual environment in which you can write advanced artificial life simulations with little effort [2]. BREVE is written in C++ and includes an OpenGL display engine for real-time visualization (Figure 3). It has full support for writing experiments in the Python programming language, making it easy to script simple control behavior or more advanced methods that requires the usage of OOP paradigm. Among its features we highlight: (1) support for collision detection with arbitrary shapes; (2) allows to build articulated bodies; (3) advanced physical simulation.

There are two basic objects in BREVE, Stationary and Mobile. The former represents immobile objects, such as a floor or obstacles while the latter represents agents which move and behave according to its own controller. At each iteration step the Control class calls the iterate method that is responsible for updating the state of Mobile objects.

![Figure 2. Mutation leads to complexification.](image2)

![Figure 3. Sample Breve screenshots.](image3)

¹NEAT Python: code.google.com/p/neat-python
Optionally you can run BREVE in background mode without a graphical interface, maximizing CPU usage and allowing to control simulations over a Linux network.

4. Simulation

Our interest in ALife is the simulation of open-ended evolution, that is, one with no restrictions regarding the topology and size of the neural network. In a typical experiment setting the individual (agent) perceives and acts in the environment through its sensors and actuators respectively. A simple modeled agent and its perception attributes is depicted at the left of Figure 4. At the right is a typical scenario where the individual has to collect the closest food it is sensing. The virtual environment is created and managed by the BREVE Control class. At each time step the neural network receives as input the value its sensors is reading and the action is taken after activating all the neurons. This process runs in loop for a fixed number of steps and evaluates the agent’s fitness based on some criteria. In this simple example there is only one input: the angle between the agent’s heading to the closest food. There are two outputs that controls how fast it moves forward and turns to the right or to the left. As is the case with genetic algorithms, the selection method varies according to the experiment where one wishes to study, e.g., gathering food, predator-prey chase, wandering behavior, and so on.

At the virtual environment is created the individual is allowed to live for a fixed number of time-steps (Figure 4) and its fitness value is calculated based on some defined criteria. When the simulation time ceases, BREVE returns the fitness for that particular individual. The process can also be easily extended to evaluate the whole population at once depending on the experiment’s requirement.

5. Future Work and Final Considerations

It was shown how to integrate a powerful neuroevolution method with a general 3D environment simulator, opening several possibilities regarding its use in Artificial Life. The next step will help us to investigate the role of adaptive behavior and learning in respect to the interactions with the environment.

From the technical point of view it would be of great value to have a parallel and distributed version of NEAT so that the evaluation of the population could be carried out in a cluster. In addition, the current version of Breve allows only one camera view at a time, but this restriction will be solved in a future release, increasing the visualizing immersion through the use of several displays. So far BREVE has shown to be very stable in long-time running simulations, becoming well suited for general purpose experiments in Artificial Life and Intelligence, eliminating the need to write a full featured simulator from scratch.

6. References
