EvolGL: Life in a Pond.

Santi Garcia Carbajal[1], Martin Bosque Moran[2] and Fermin Gonzalez Martinez[2]

[1]Computer Science Department, Unuiversity of Oviedo, Spain
[2]INDRA Graphics Department, Alcala de Henares, Madrid, Spain

Abstract

In this work we present the first version of Evolgl, an artificial environment for the development and study of 3D artificial lifeforms. In this first phase on the development of the project we have focused in setting up a virtual world governed by its own laws, whose state had direct influence upon the artificial beings that inhabit it. Starting from the definition of this virtual world, we have designed a basic type of creature (Evolworm), and the genetic coding of its main characteristics. Evolutionary techniques are then used to evolve the morphological features and behavioral aspects of Evolworms. They must learn to be unfolded inside the world, escape from their enemies, find couple, and obtain food. All of this in absence of an explicitly defined fitness function. In the future we are using this environment to study some classical techniques in the evolutionary computation field, like niche programming, and promotion of junk code (introns). GA-P techniques are used to code the external appearance of the individuals (the texture), to let evolution end up with individuals adapted to be invisible in some zones of the world. The artificial system of vision, and the implementation of the worms' behavioral mechanisms so that their actions are provoked exclusively by the sensory information are still under development. At this moment, we have obtained distinct forms of evolworms, as well as different bosses of behavior that we describe in this article.

Introduction. Related work. GA-P algorithms.

In this section we review some existing works from different authors that have worked in similar artificial life projects. Most of them focus in obtaining highly realistic graphical results, or in the development of an accurate physic model of the environment. We have discarded the development of a physics-based virtual marine world, due to the high computational cost of such a model, to focus our work in the behavioral aspects of the evolution, and deserve some calculation possibilities to enhance the artificial vision system of the creatures. We include also a brief explanation of GA-P algorithms, an hybrid between Genetic Programming and Genetic Algorithms that we have used to code some features of the artificial creatures that live in Evolgl.

Related work.

M. Komosinski and Szymon Ulatowsky (1998) have developed during the last years the *Framsticks Project*, where some artificial living organisms called *Framsticks* learn to maximize any given criteria (total displacement over the environment, speed, etc). Framsticks are constructed from sticks, three kinds of receptors, muscles, and a neural controller for the resulting body. These authors use a coevolutive approach to obtain concurrently the best adapted body and the optimal controller in order to reach a pre-stated goal. Complete information about this project is maintained at `http://www.frams.poznan.pl/`. Karl Sims' works are broadly known in the evolutionary community. We have mainly looked at *Blockies*(1992), because of their ability to evolve complex behaviors from a numerical fitness function. Despite their visual realism, Blockies are not thought to be built, becoming real objects. Since the evolutionary computation process generates both the physical architecture and the control programs, these works fall into the body-brain coevolution category, like Framsticks. Xiaoyuan Tu and D. Terzopoulos (1994, 1995) have developed a realistic model of a virtual marine world, where three kinds of fishes swim and interact. The result of their work is very realistic. The mental state and habits of the artificial fishes are modeled through three mental state variables (Hunger, Libido, Fear), and habit parameters. An intention generator choses a behavior routine from the sensory information. The implementation described runs on a Silicon Graphics R4400 Indigo2, with 10 fishes, 15 food particles, and 5 static obstacles, obtaining 4 frames/sec. Our hardware is very modest, and we need to work with populations of great size to expect that the evolution take place. Therefore, we carried out a simulation of a very simplified marine world. Technosphere is an internet-based artificial life simulator allowing people to create their own creatures, and receive communications from them as they grow, evolve, and die in a 3d virtual environment. Users create artificial lifeforms and follow their interactions in the virtual environment, via e-mail and a suite of world wide web based interface tools. Full information about this project is available at `http://www.technosphere.org.uk/`

GA-P algorithms.

GA-P technique (Howard and D'Angelo, 1995) is an hybrid between genetic algorithms and genetic programming, that was first used in symbolic regression problems. Individuals in GA-P have two parts: a tree based representation and a set of numerical parameters. Different from canonical GP, the terminal nodes of the tree never store numbers but linguistic identifiers that are pointers to the chain of numbers (see figure .) The behavior of the GA-P algorithm is mainly due to its crossover operator. Either both parts of the individual can be selected and crossed. We have employed GA-P algorithms in the identification and control of complex dynamical processes, and in classification problems (Garcia and Gonzalez, 1999). In this work, we use the GA-P approach to code and evolve some of the features of the living beings that inhabit evolgl. The most important feature we have modeled with GA-P is the Texture Composition System, to obtain the final texture of each individual from a composition of the basic textures. We discuss this point in section

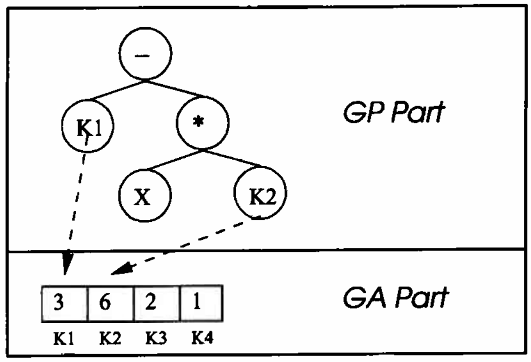

Figure 1: Representation of a generic individual in GA-P techniques. GA-P individuals consist of two parts: a tree and set of numerical constants.

Why Evolgl?

As we previously mentioned, there is a number of works where some kind of artificial creature is modeled, with extremely realistic results, like Artificial Fishes, or Framsticks. Other works obtain artistically interesting results, like some works of Karl Sims. Unfortunately, accurate mechanics, a complex hydrodynamic model, and large population of artificial creatures are not possible when genetic operators (selection, crossover, mutation) are not implemented in the standard way in Evolutionary Computation Techniques. In an environment like Evolgl, where generations do not exist, and the fitness function is not explicitly defined, the main attention and computational effort is dedicated to the behavioral aspects of evolution. Our aim is to obtain, in the next phases of this project, an artificial life simulator where we can test some classical techniques in the evolutionary computation field, like niche programming, or observe the effects on evolution of the absence or presence of junk code in the internal representation of the individuals. There is no spectacular result from this work. Evolgl is ready now, and we plan to include the stereoscopic vision system in the next months, to test the importance of the Texture Combination System we have developed for the Genotype-Phenotype mapping process.

Evolgl.

The world.

Evolgl is a virtual world consisting of two zones: the exterior is a forest in which life takes place in a pond. The function of the forest is strictly aesthetic. All the events, calculations, the evolution itself, occurs below the surface of the pond. It is limited by four rocky walls, and warmed up by the action of four light sources (red, yellow, white, and green) with different pulses. See figure 2. The state of each light source causes the apparition of vegetation in greater or smaller measurement in its neighborhood. The plants generate directly the balls of energy Evolworms are fed with. This way, at every point of the pond, there is a temperature and a vegetation density. Plants are born and die as the nearest light source oscillates in its intensity. We introduce the different pulses and intensity for each light source to simulate periodicity in the virtual world and force Evolworms to develop the ability to store enough food to survive the "cold" periods. Moreover,we made the consumption of energy of Evolworms to depend on the temperature of the environment. Figure 3(left) shows the external aspect of the pond, and a view taken from into the pond (right). Figure 2 (right) shows a plot of the wire frame model of Evolgl, and the situation of the four light sources.

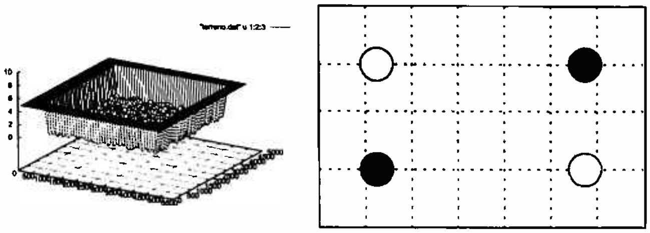

Figure 2: Evolgl. a plot of the wireframe model of the pond (left). Situation of the four light sources(right).

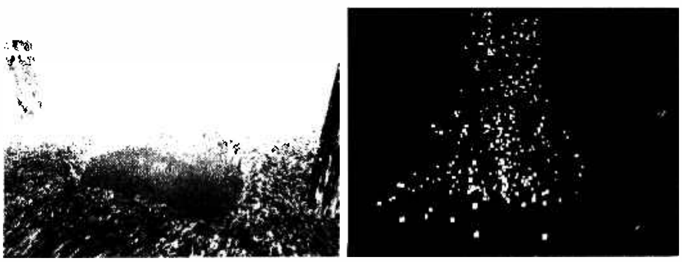

Figure 3: External view of the pond (left). A view from inside the pond (right).

Evolworms.

Evolworms are the artificial living organisms that inhabit Evolgl. We choose this name because of their appearance, that reminds the one of some kinds of real aquatic worms. In this section we describe the morphological, behavioral and aesthetic aspects of this creatures, and explain the genotypic representation we created to simulate the evolutive process in the pond.

Physical structure and appearance. Figure 4 shows the generic structure of an Evolworm. Figure 5 shows some Evolworms in wireframe model.

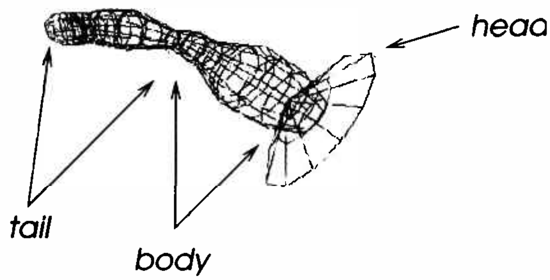

Figure 4: An Evolworm. There is three components: head, body, and tail.

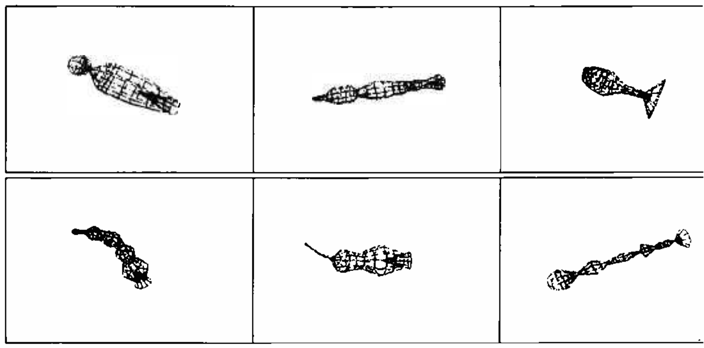

Figure 5: Evolworms. Wire frame model.

There is three different components:

1. Head. The shape and size of the head determines the vision angle of the worm, and the penetration factor into the water. A creature with a big head will have a good perception of the environment, but will move slower than the same individual with a thinner head.

2. Body. The volume of the body determines the storage ability of an individual. This feature lets some Evolworms to be less dependent of the stational variations of the environment than other. Of course, an excessive growth of this part will lead to slow individuals, easier to hunt.

3. Tail. The number of segments of the tail, and their lengths determine the self-protection, and attack abilities of Evolworms, since the way to kill another individual is by hitting him repeatedly with the tail. Besides, the consumption of energy per time unit is partially determined by the total volume of the individual.

The "brain". The transition from one state of the automata to another is partially determined by the weight of the arc that connects both states. This way, over a generic representation of the automata, many different kinds of individuals (aggressive, coward, food seekers) can exist. Some transitions are possible only under certain conditions. For example, an evolworm will never attack another if its color says "I am an enemy". The list of enemies is used to escape from possible predators, and to identify food (energy balls, or other individuals). It will evolve to store the best set of colors in order to survive into Evolgl. Figure 6 shows the behavior automata of an Evolworm.

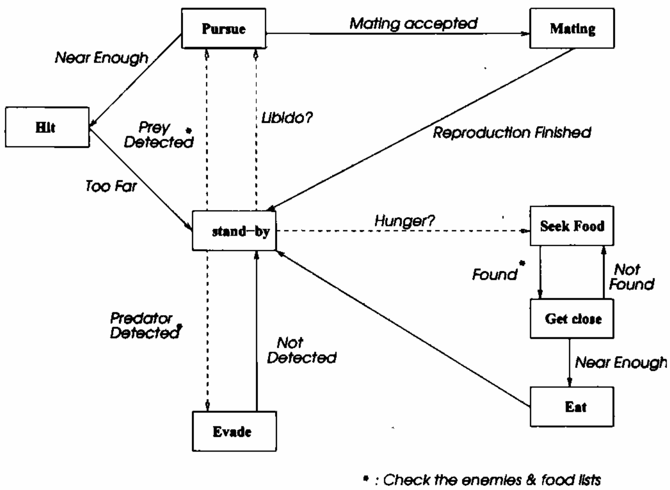

Figure 6: The brain (simplified). Discontinuous traces mean transitions that are executed probabilistic.

Genotype. Figure 7 shows the basic genotypic representation of Evolworms. It consists of the following elements:

1. The behavior automata (the "brain" of an individual).
2. A list where the individual stores the colors of his enemies.
3. A list containing the colors of any kind of food (energy balls, other individuals)
4. The structural features of the individual: number of segments, radius of each segment, etc.
5. The genetic code for the vision system.
6. Additional features like color detection tolerances, re-activity, etc.
7. A GA-P expression that codes the external appearance of the "skin".

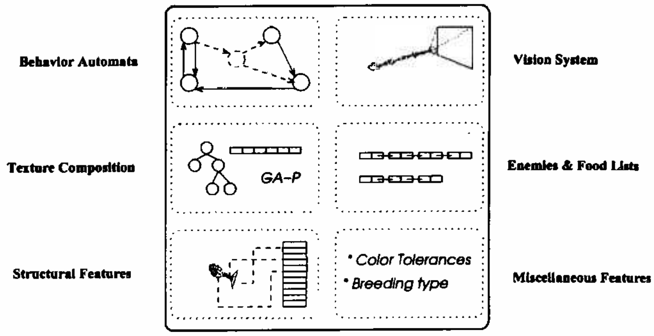

Figure 7: Evolworms: genotype codes the structural and behavioral features of each individual.

Vision system. In this first version of Evolgl, the vision system is quite simplified. Due to the computational costs of keeping large populations of evolworms wandering around, fighting and mating, we had to make a simple and static vision system. Evolworms have a viewing range and angle of view. These characteristics are part of genotype. From the 2D projection of what an Evolworm can see at each moment, it decides if there is another individual detected, food, or anything. To do this, the individual musts check the enemies and food lists, and use the tolerance parameters coded in the genotype to decide if an object is an enemy, food, or nothing relevant. See figure 8. To date, a more complex vision system is under development. Briefly, the idea is to introduce in the genotype a GP evolved function that receives as input information from an standard stereoscopic vision system, and returns a vector of three components that determines the advance direction of the individual, together with an indication of the kind of object detected.

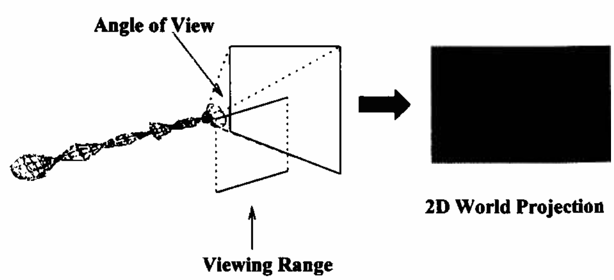

Figure 8: Artificial vision system.

Genotype-Phenotype mapping : The Texture Combination System. Most of the structural features stored in the genome (number of segments, radius, vision angle, etc) can be translated to their visual representation in a straightforward manner. We explain in detail the method we use to code and evolve the external texture applied on the wire frame model of evolworms, because we think it will be very important in the evolutive process when the vision system we are using now will be replaced with a more complex one. The coding of the texture in a way that let the evolution process end up with creatures adapted to be "invisible" in some zones of the pond is based in a GA-P representation (Howard and D'Angelo, 1995) of the external texture of an individual. We adapt the basic GA-P scheme by keeping a structural part (GP Part) that represents a set of operations made on and between different basic textures. The GA part stores the set of textures for an individual. We adapted the genetic operators to work over such a scheme. GP crossover and mutation are the standard ones defined by Koza (1994) . GA crossover and mutation are adapted to perform recombination between 64 per 64 pixel images instead of between numerical parameters. See figure 9.

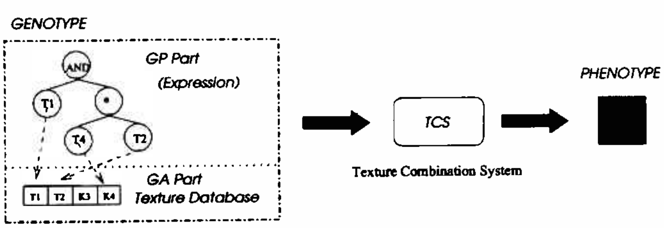

Figure 9: GA-P representation of the texture of an individual. Texture Combination System maps an expression into a resulting texture.

Experiments.

Hardware settings.

We have used a 24-node cluster based on MOSIX (Barak et al., 1999) to carry on the evolution process. All the machines included in the cluster run Red Hat 7.2. when the evolution process is quite advanced, Evolgl is visualized on a intel-based Pc running W98 with a ELSA Gladiac MX graphics card. We chose this configuration because of the poor performance of this card under Linux.

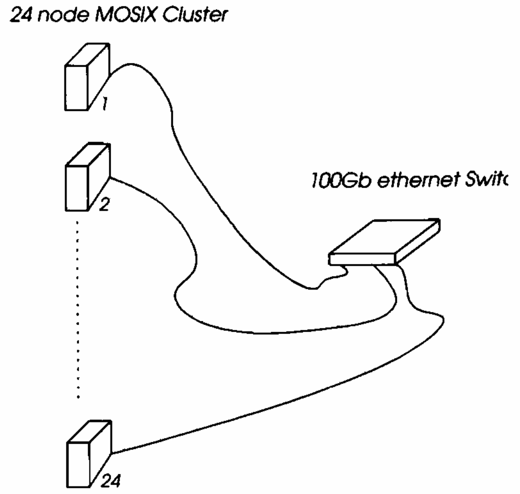

Figure 10: MOSIX cluster.

Niche Programming Model.

Results.

We distinguished three kinds of individuals with respect of the kind of food they are able to eat (energy balls, another individuals, both things), and named each specie Herbivorous, Carnivorous, and Omnivorous. We restricted the mutation operator to not let an individual change his specie, and crossover is permitted only between individuals of the same specie. The aim of these experiments was test and tune the ranges for all the parameters involved in the evolution process, check the maximum number of individuals Evolgl was able to deal with, and debug all the components of the system. Table 1 resumes the parameters of each run. We observed that the environment can manage over 600 individuals, so we restricted the experiments to work with accumulative populations always under this number.

Table 1: Evolgl. Preliminary results.

Experiment	Species	Individuals
A	Herb. + Carn.	200 + 200
B	Herb. + Omn.	200 + 200
C	Carn. + Omn.	200 + 200
D	H. + O. + C.	150 +150 + 150

Figure 11 shows some Evolworms obtained after leaving the environment evolve during a period of 24 hours.

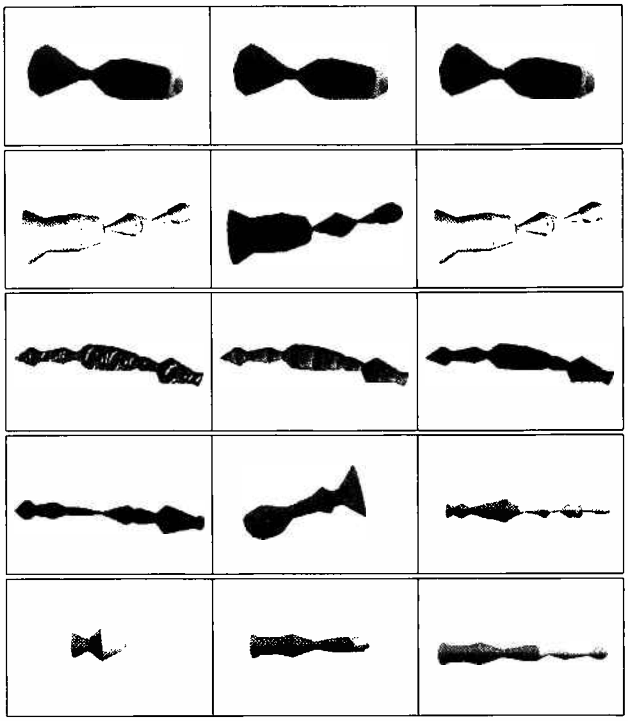

Figure 11: Some pictures from Evolworms.

Conclusions and future work.

We have given the first step toward the development of an artificial world were living organism can evolve and learn complex behaviors. There is still a lot of work to do, because we hope that the final model of evolworms will comprise:

1. An artificial vision system based on the subjective view of the world that an individual has, that provokes directly changes on the behavior automata, and generates the actions of the individuals.

2. A set of functions genetically induced that generate the movement of each part of a creature only from a set of *sensations* like temperature, oscillations produced by other creatures, etc.

References

[1] Barak, A., La'adan, O., and Shiloh, A. (1999). Scalable cluster computing with mosix for linux.

[2] Garcia, S. and Gonzalez, F. (1999). Evolving fuzzy rule based classifiers with GAP: A grammatical approach. In Poli, R., Nordin, P., Langdon, W. B., and Fogarty, T. C., editors, *Genetic Programming, Proceedings of EuroGP'99*, volume 1598 of *LNCS*, pages 203–210, Goteborg, Sweden. Springer-Verlag.

[3] Howard, L. M. and D'Angelo, D. J. (1995). The GA–P: A genetic algorithm and genetic programming hybrid. *IEEE Expert*, 10(3):11–15.

[4] Komosinski, M. and Ulatowski, S. (1998). Framsticks - artificial life.

[5] Komosinski, M. and Ulatowski, S. (1999). Framsticks: Towards a simulation of a nature-like world, creatures and evolution. In *European Conference on Artificial Life*, pages 261–265.

[6] Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA.

[7] Sims, K. (1991a). Artificial evolution for computer graphics. Technical Report TR-185, Thinking Machines Corporation.

[8] Sims, K. (1991b). Artificial evolution for computer graphics. *ACM Computer Graphics*, 25(4):319–328. SIGGRAPH '91 Proceedings.

[9] Sims, K. (1992a). Interactive evolution of dynamical systems. In Varela, F. J. and Bourgine, P., editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 171–178, Paris, France. MIT Press.

[10] Sims, K. (1992b). Interactive evolution of equations for procedural models. In *Proceedings of IMAGINA conference, Monte Carlo, January 29-31, 1992.*

[11] Sims, K. (1993a). Evolving images. Lecture. Lecture presented at Centre George Pompidou, Paris on March 4, 1993. Notebook. Number 5.

[12] Sims, K. (1993b). Interactive evolution of equations for procedural models. *The Visual Computer*, 9:466–476.

[13] Terzopoulos, D. and Rabie, T. (1995). Animat vision: Active vision in artificial animals.

[14] Terzopoulos, D. and Tu, X. (1994). Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, 1(4):327–351.