

A New Step for Artificial Creatures

Nicolas Lassabe
IRIT/UT1 UMR 5505
2 Rue du doyen Gabriel Marty
31042 Toulouse cedex 9
FRANCE
Email: lassabe@irit.fr

Hervé Luga
IRIT/UT1 UMR 5505
2 Rue du doyen Gabriel Marty
31042 Toulouse cedex 9
FRANCE
Email: luga@univ-tlse1.fr

Yves Duthen
IRIT/UT1 UMR 5505
2 Rue du doyen Gabriel Marty
31042 Toulouse cedex 9
FRANCE
Email: yves.duthen@univ-tlse1.fr

Abstract—In this article, we present our research on the evolution of morphology and behavior of complex creatures in virtual environments. We propose to study the evolution of creatures faced with different situations from crawling and walking to more complex activities like climbing and skating. Our creatures use solid 3D blocks and *graphtals* like Karl Sims's creatures and a new type of controller based on classifier system. The results constitute a new step toward creatures adapted to more complex environments.

I. INTRODUCTION

In this article we propose a new step in virtual creature evolution. In recent years, many articles on evolving creatures have been released [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. They often focus on the morphology and behavior of one or two creatures but do not take into account the complexity of their environment. In contrast, some simulations focus on the interactions of population dynamics but deal only with much simpler creatures [13], [14]. Our approach tries to merge both approaches to evolve complete complex creatures in complex environment. After a brief discussion on previous approaches, we will present our system and focus on a new type of controller based on classifier systems. We will then show how evolved creatures are able to cope with various situations caused by a complex environment.

The present work is organised as follows: in the next section will present the works about evolving creatures. In the third section of this article we will present a brief description of our creatures. We will particularly emphasize our proposition to use a classifier system as a tool to model complex activation function. We will then present a set of benchmark tests to evaluate our architecture and we will describe the results. At the end we discuss how various fitness functions have an influence on the phylogeny of creatures and on various ideas for future studies. In the results section, we present the best evolving creatures from our experiments. The study of emerged strategies to resolve the various problems is promising and opens new perspectives. We discuss the influence of fitness on the symmetry and modularity of creatures. To conclude, we present some perspectives of the project and the interest that a complex ecosystem environment presents.

II. RELATED WORK

There are many existing solutions for representing morphologies and controllers. One of the first works was published

by A. Fukunaga, J. Marks and J.T. Ngo [15]. Karl Sims's creatures certainly remain the most evolved [8], [9]. He has achieved evolutions bound by very few constraints and obtained a large diversity of creatures giving birth to crawling and swimming behaviors. Thereafter many works reproduced his results with different approaches for the representation of morphologies and controllers, such as the L-System creatures of Hornby [3] and Lipson's Golem project [5]. Their goals were to optimize 3D creatures in relation to a preset fitness inside a physics engine and to reproduce them in the real world. Another approach to obtain more complex behavior is to optimize a set of 2D creatures in a virtual pool by means of natural selection [13], [14]. Each creature has an amount of energy and tries to survive looking for food in the environment. Dealing with simpler creatures, this approach allows to simulate many creatures at one time. Komosinski did the same experiment in a 3D environment [4]. Recently, the *EvoGL* project [16] proposed to simulate a virtual world with worms. The evolving worms belonged to herbivorous, carnivorous and omnivorous species and developed different strategies of survival.

These works distinguish two distinct ways to optimize complex behavior evolving creatures: with a preset fitness or by means of natural selection. The preset fitness approaches continue to be improved with recent results, like evolving flying creatures [7] and evolving creatures for fighting [6]. In these works, Miconi reproduced the Karl Sims's creatures and improved the co-evolution [17], [18], [12]. One promising perspective for the morphology of creatures could be artificial embryogeny to have a more complex representation of creatures [19]. However, much could be improved to offer better results in more complex environment [20], [21], [22]. The works based on natural selection are becoming achievable and will certainly be interesting to have more adaptive solutions [16].

III. ENVIRONMENT AND ARCHITECTURE

A. Morphology

In present work, our creatures use a morphology based on *graphtals* similar to those from Karl Sims's ones [8], [9]. The genotype represents the creature's development. For each creature we define a root node. Each node contains the information of one block. Blocks are represented by solid 3D

cubes. The data of cubes are their x , y and z sizes, color and the link to the parent block if it is not the root block. The link between two nodes will be represented by a type of joint in the phenotype. A block could be linked with itself, this providing the possibility of repeating the expression of a node. The modularity is very visible in our results. We choose not to explicitly represent the symmetry in our creatures as in Karl Sims to study how the evolution could make it appear.

This morphology is not static but could be extended because we want to keep the ability of generating various lifeform inside our futur ecosystem. It provides us with good results for our current simulations (Fig. 12) but we plan in the near future to broaden the types of joints between blocks and types of blocks defining the creatures's morphologies.

B. Controller

The creature uses a classifier system [23], [24] as a controller. The controller needs to activate the articulations correctly (junction between the blocs). The requested signal to do this needs to be continuous and must vary in time according to the incoming information issued from the sensors. This signal is generally a sine wave [4] or a composition of periodical functions but it is also possible to compose a signal from some preset patterns (Fig. 1). In this latter approach, the main problem is to conserve the continuity of the signal, while selecting the next adapted pattern. We start by defining a data base of one thousand random generated patterns. We then define one main composition rule: two patterns could be appended to compose a part of the control signal only if the following conditions are satisfied:

- Patterns are defined between $[-1,1]$ and the same holds for their minimum and maximum.
- If P_1 and P_2 are two patterns and P_2 is appended to P_1 , it is necessary that:

$$(P_1(1) - P_2(-1)) < \varepsilon$$

where $\varepsilon = 0.01$

The classifiers control the selection of patterns. The entries of the classifiers, which define the incoming information from the environment, are the creature sensors. According to the sensors's state, a set of rules pointing to patterns in the data base is triggered. The composed signal is used by the effectors controlling the creatures's muscles. If the resulting action is satisfactory, the rules used to generate that action are rewarded, otherwise they are penalized. The advantages of this method lie in the fact that it is easy to enrich the used data base and obtain more complex signals than those obtained by simple primitive signal composition.

Another approach could be to use mathematical functions (which may be generated by the use of Genetic Programming [25] for example) to replace the pattern. However, there are too many possibilities and we do not exploit all of them here.

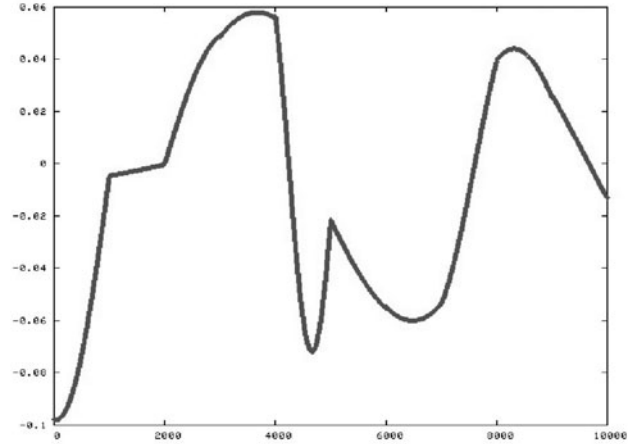


Fig. 1. Signal composed by a classifier system for a controller (x axis: steps, y axis: velocity).

C. Genetic operator

1) *Mutation*: The mutations are operated on the graph, hence if a mutation is operated on a node recursively reproduced, this mutation will affect all the blocks issued from this node. This allows our structure to exhibit modular properties such a pattern repetition.

The possible mutations on the graphtal:

- A new block could be added randomly to the graph.
- A block could be deleted randomly from the graphtal. It is necessary to check whether or not this operation is possible. This operation cannot be applied to the root block. A part of the subgraph could be deleted by this mutation if blocks are connected to the deleted block.
- A mutation could slightly and randomly move one of the axes x , y or z of the parent block link between two connected blocks.
- A mutation could slightly and randomly move one of the axes x , y or z of the link between two connected blocks.
- A mutation could slightly and randomly center one of the axes x , y or z of the between one block and its parent.
- A mutation could slightly and randomly center one of the axes x , y or z of the link one parent and one of these blocks.
- The size x or y or z of a randomly selected block can be randomly changed.
- The number of iterations of a random node can be randomly changed.
- The number of iterations of a random node can be increased by one.
- The number of iterations of a random node can be decreased by one.
- The axis of a random joint can be changed.

2) *Crossover*: We use the *graphits* method as in Karl Sims's works [8], [9] to reproduce two creatures. That is, we randomly select one node in the graph of the first creature parent and another one in the second parent. To finish, we permute the

subgraphs issued from the selected nodes. This approach is simple and gives us good results. We plan to implement the other method by Karl Sims, *crossover*, the *picking* method of Miconi [18] in the next step of our work and compare the results to see the influence of the different methods of crossing over.

3) *Genetic algorithm*: The population of the simulation is initialized to one hundred individuals. Each creature's genotype is randomly created according to a maximum number of blocks. For a typical simulation, this number is generally set to ten. This number could quickly increase by mutation of the number of iterations on each node provided by the modular definition of our genotype. Each individual's performance is then evaluated in the simulated environment during 40000 steps (40s). The fitness used to evaluate the creatures will be described later for each experiment. For the selection process, we use tournament selection between three individuals. The rank of reproduction is seventy percent and that rank of mutation is ten percent. The evolution stops when the desired fitness is attained.

D. Breve

The implementation of a physics engine is a difficult task requiring a great deal of work. Our choice was therefore to use Breve [26], an engine based on ODE (Open Dynamic Engine). This environment is well suited for artificial life simulation because it provides a set of high level primitives which suit the simulation of articulated bodies. Breve includes low level physics dealing with torques and strength computation, gravity, collision detection but also several higher level property computations. For example, Breve already implements a lot of useful functionalities such as the quantity of light perceived by an object that we plan to use in future experiments. Our program is written in C++ and inserted in Breve as plugins. In this way, our program is not dependent on Breve and can be used as a library with another physics engine.

IV. RESULTS

We here show the ability of our system to evolve creatures in different situations. In general, in our experiments, the fitness tries to maximize the distance traveled by a creature. All the creatures are recorded on videos¹.

A. Walking on a flat floor

The first experiment is the traditional walk. This original experiment is similar to the work of Karl Sims [8], who presented the first one. Many works try to reproduce it. Its goal is to optimize both the morphology and the controller of a creature which has to walk on a flat floor environment.

In this experiment, our fitness F is not the distance traveled d but the average speed during the time t of evaluation of the creature. $F = d/t$

One of the best results that we get is a tripod (Fig. 2.a). This creature is not symmetric: it has only three blocks for

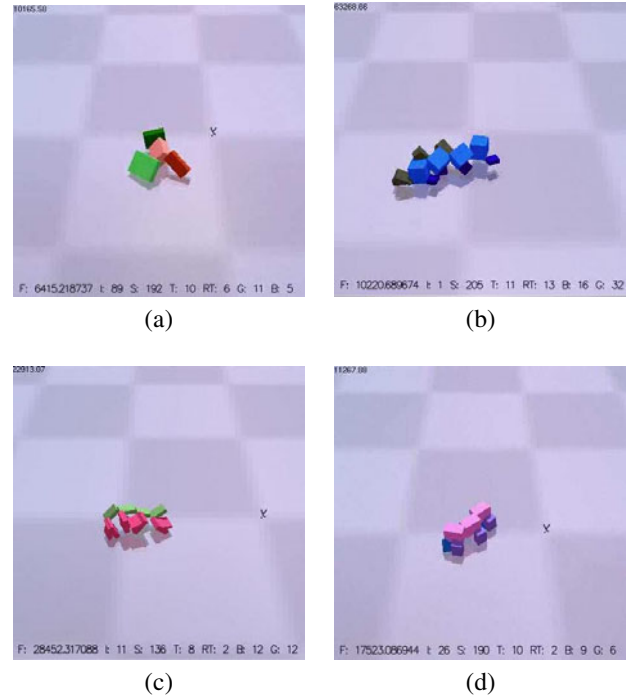


Fig. 2. (a) Evolving creature walking like a tripod. (b) Evolving creature walking like a crab. (c)(d) Creatures with a modular body

legs and one for the body and is not modular (no repeated block). Various other creatures used interesting strategies to move like a crab or have kinds of legs (Figs. 2.(b-d)) or rolling (Fig. 3.(a-b)).

B. Walking in a specific direction

In the experiment, we keep the same simulation and parameters but we change the fitness function. In this case, we indicate a specific direction for the creature, for example, to follow the x axis. The new fitness is $F = |x|/t$

One might imagine the results would be the same. In fact, the creatures are optimized to go in this direction and this is manifested by a perfect symmetry of the best creatures (Fig. 4). An interesting feature of our creatures' morphology is that they are composed of a modular body. This modularity gives them a great efficiency. Among the best creatures are kinds of worms with some legs allows them to walk with long steps (Fig. 4(b-c)).

C. Thought trenches

In these experiments, we isolate the creature on a small cube. Some other cubes are aligned with the first one separated by trenches. The creature's goal is to join the last cube. Many strategies are possible like jumping over trenches. Or, if the creature is tall enough; it could crawl to the next cube.

1) *Interest of the experiment*: The interest of this experiment is that this situation often appears in reality and is a difficult task to resolve. It is easy to improve the difficulty of this task changing the width of the trenches. It is possible to alternate the positions and heights of cubes.

¹A video with the best creatures is on the first authors's home page: <http://www.irit.fr/~Nicolas.Lassabe/videos/alife2007.mov>

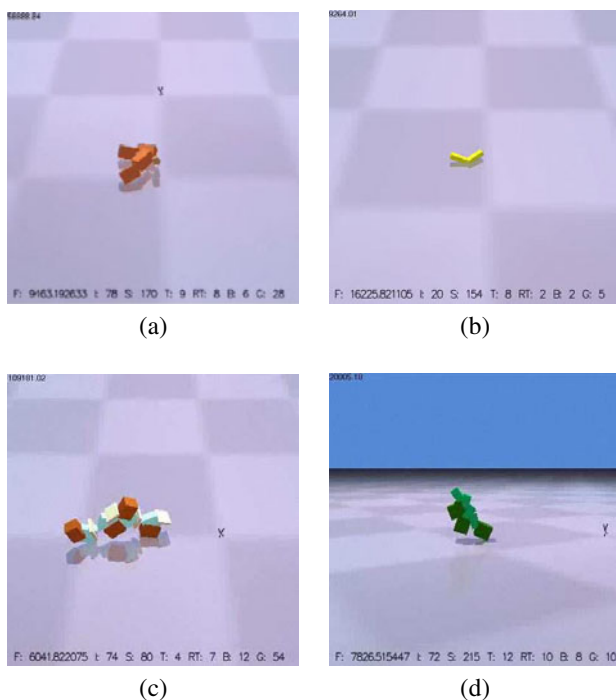


Fig. 3. (a)(b) Evolving creatures rolling. (c) Look like a robot (d) Creature with a human profil

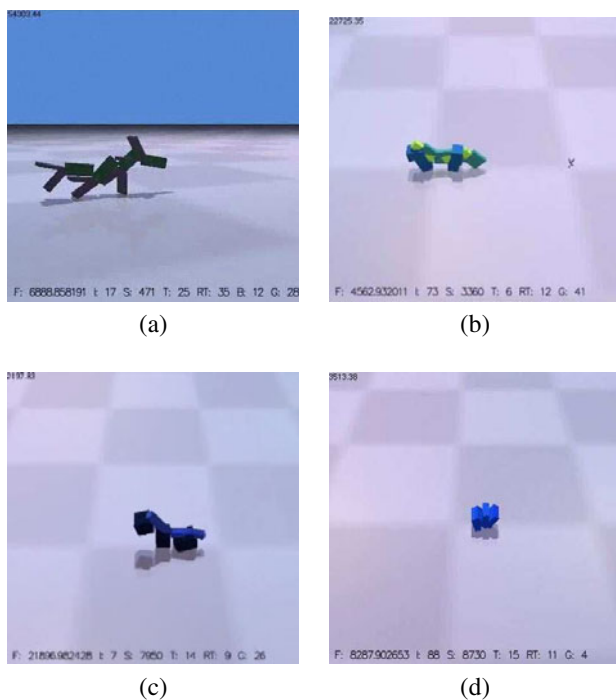


Fig. 4. (a) Evolving symmetric creature walking in a specific direction are symmetric. (b)(c) Modular evolving symmetric creatures with kinds of Legs. (d) Creature Roolling

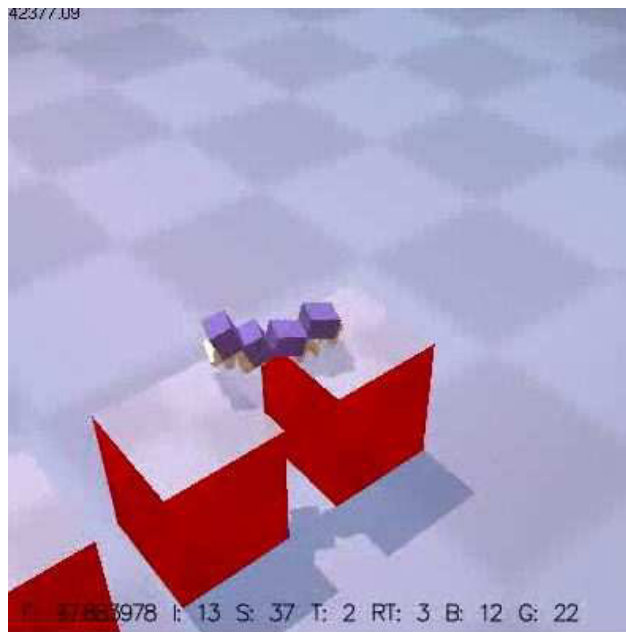


Fig. 5. Evolving creature walking cube by cube

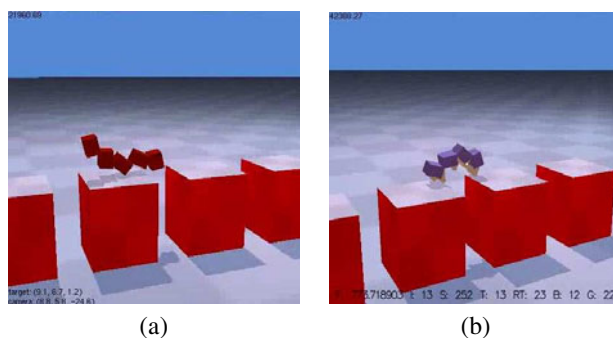


Fig. 6. Two evolving creatures on cubes.

2) *Experiment:* In the first simulation, the trenches are not very large, just a quarter the width of the cube. The fitness is defined to attract the creatures on the last cube. If the creature falls down the fitness is divided by ten, to push the creature to go in the direction of the next cube. It should not be possible for a creature to mount on top of a cube. The best creature uses its legs to cross the holes (Fig. 5.(b)).

D. Stair Climbing

The environment of the creature is often a flat floor. To increase the difficulty, we try to see if an evolution could generate a creature able to climb a stair. The difficulty could be increased by the height of steps. In the first test, we set up the height of the steps to half the average size of creatures and the width as the average width of the creature. The fitness is defined to indicate the direction to the top of the stair and optimizes by speed to which the creatures go.

1) *Interest of the experiment:* Stairs are everywhere in daily life, and can be of several different types. It is a difficulty for a

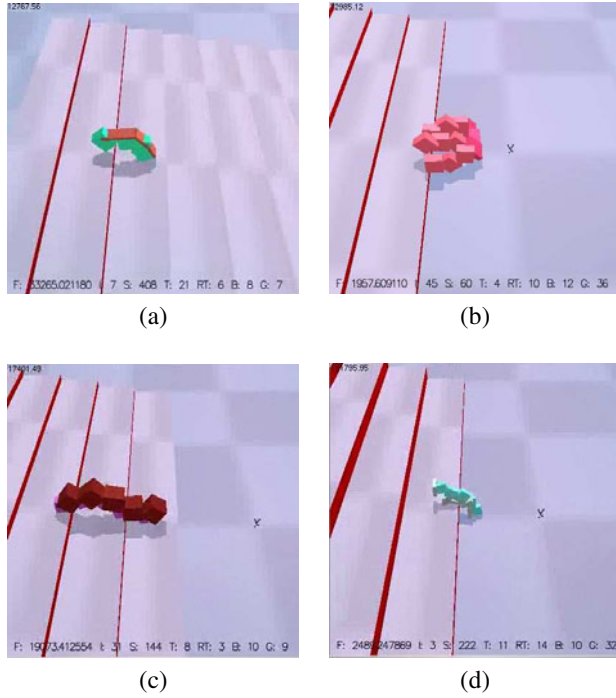


Fig. 7. (a) Evolving creature jumping step by step. (b) Evolving creature climbing a stair. The body of the creature is modular with optimized legs to quickly climb the stair. (c) Looks like a worm, (d) looks like a hand.

robot. Using evolving creatures to see the different strategies to climb a stair is an interesting approach. The morphology, the balance and the synchronisation of movements are difficult to design to work on any stair [27].

2) *Experiment*: The best creature can climb to the top of a stair quickly. The strategies employed are different. Some creatures which move like a worm are able to climb almost two steps at a time (Fig. 7.(c)). Others are perfectly designed to jump step-by-step (Fig. 7.(a)). Once again, the bodies of the best creatures are modular. Moreover the creatures are also symmetric (Fig. 8). If a creature has a good potential to climb but is not totally symmetric, the evolution will optimize the creature in this way. To improve the difficulty of the experiment, we increase the height of the step and run a new evolution. The best creature uses a new strategy to climb the step using flat legs (Fig. 7.(d)).

E. Walking terrain

The world is rarely perfect, so, in this experiment, we design a terrain with hills and dales. The amplitude of these heights is ten times greater than the average size of the creatures. The fitness is the distance traveled divided by the time of the simulation as in the walk experiment: $F = d/t$. Our creatures have more difficulty in being effective in this type of environment. This is because the environment is unpredictable. The best results are less efficient than the creatures on the floor but still have good performance (Fig. 9).

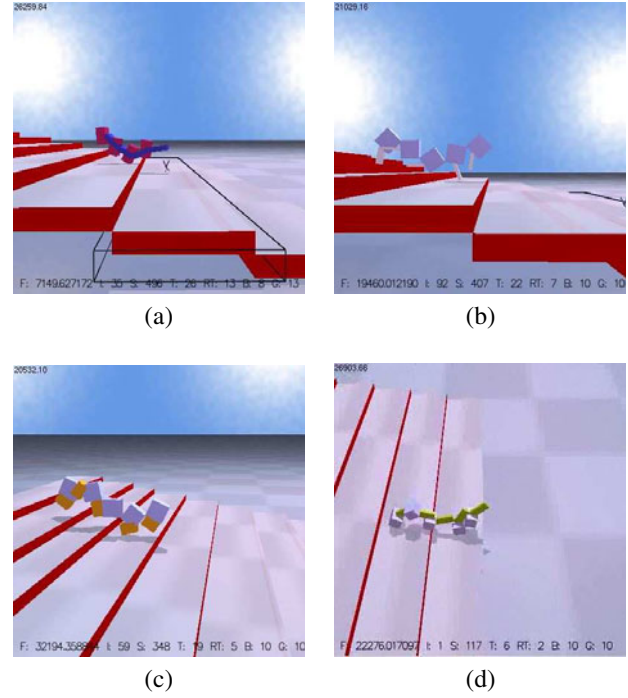


Fig. 8. Evolving creatures climbing a stair. The body of each creature is modular with optimized legs to climb the stair quickly.

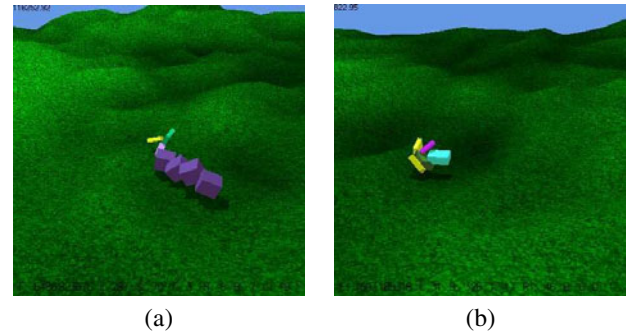


Fig. 9. Evolving creatures crawling on a terrain.

F. Skating

The interaction within the environment is an important issue in improving the complexity of evolving creatures. In general, we underestimate the capacity and the power of adaptivity of evolving creatures. Many experiments need to be tried to find out their real limit and their potentialities. In this experiment, we define a kind of skateboard composed of four wheels and one large board. The goal is to see creatures could use it whether to move in some direction. For this, the creatures fall down close to the skateboard. The creature and the skateboard are two different objects and do not have any connection between them. The function fitness is defined to maximize the distance covered with skate ds .

$$F = d + ds - (d - ds)$$

Many strategies shows employed: some creatures jump on

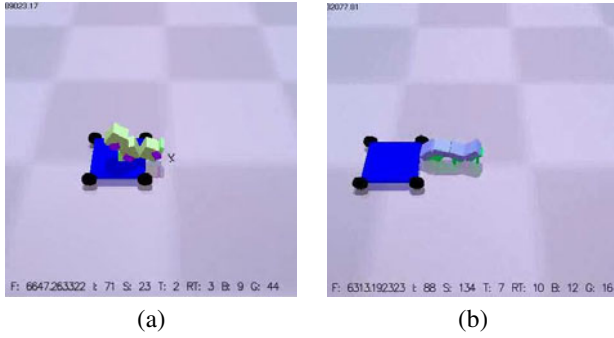


Fig. 10. (a) Evolving creature skating by jumping on the skate. (b) Evolving creature pushing the skate.

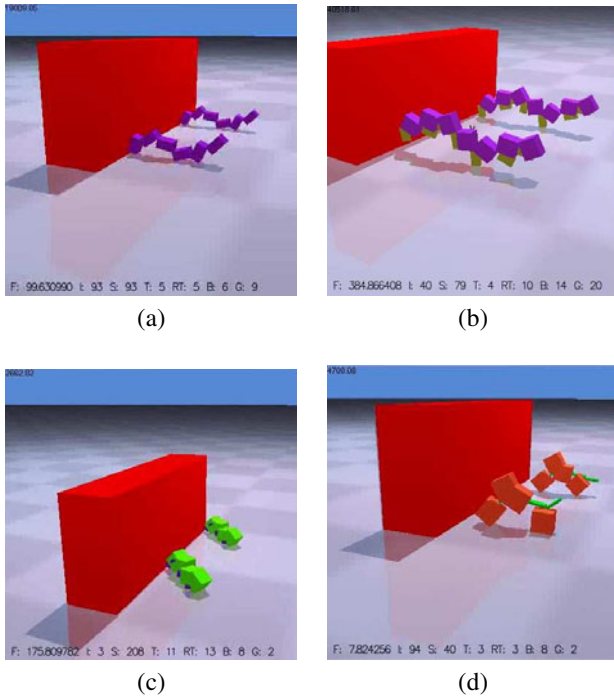


Fig. 11. (a)(b) Evolution of creatures few generations later. (c) Evolving creatures pushing a cube. (d) Cooperation between evolving creatures.

the skate to slide with it (Fig. 10.(a)), others push it and run behind it (Fig. 10.(b)) and others go up and push from the back to move. These results show the ability of the evolving creatures to be adapting to the environment.

G. Cooperation

In the last experiment, the goal is to push a block in the right direction. The block is defined to be too big for only one creature to push. For this experiment, one creature and his clone are initialized for each simulation in the environment. The fitness is to maximize the distance traveled by the wall dw (center of gravity) in one direction (axis x): $F = dw$. In this experiment, the best creatures are bigger to push the wall (Fig. 11) and give every good results. In the future, we plan to improve this experiment, using two different creatures.



Fig. 12. The project of our ecosystem

V. DISCUSSION

These results prove the possibility of using a new kind of controller based on classifier systems to control evolving creatures in various activities. The next step is to emerge our system inside our virtual ecosystem with plants and various lifeforms. We have already started this implementation [28] (Fig. 12).

A. Implementation

For the moment, the lifeforms and most of the plants have been implemented. An evolution of the creatures with a preset fitness presented in this article give some good results. For the final simulation, we will use Breve network implementation to distribute all the calculations on different machines. The complete simulation will require a great deal of computation and will not be computed on a single machine. Every lifeform is implemented in C++ in different plugins and these plugins are interfaced with Breve. We try to maintain a maximal flexibility to add different lifeforms in order to get the broadest, most various and interesting results possible.

B. Why ecosystem?

The realization of an ecosystem can allow the generation of lifeforms able to adapt to a dynamic environment, the specification of species and the emergence of life cycles. The interest of simulating such virtual ecosystems are several. In the field of biology, it could lead to a better understanding of the evolution and extinctions of species as well as of some mechanisms of life [29]. The evolution of morphology and behavior in relation to a specific environment has an interest for the design of robots and their controllers [30], [5],

[31]. A complex environment should generate adapted survival strategies during evolutions cycles.

Interesting questions are whether different evolution scenarios can emerge from an identical initial environment and whether the environment complexity is a relevant criterion on the emergence of different survival strategies [20], [21], [22]. If that were the case, how does one evaluate complexity? What are the elements necessary for the emergence of complex life forms and behaviors? Is the use of a complex physics one of these elements or can biological complexity be supported by simpler physical models? What is the importance of the interactions? These are open questions, which may be answered through experimentation with a complex artificial ecosystems.

VI. CONCLUSION

We have shown that it is becoming possible to evolve virtual creatures in complex environment that provide us with physical simulation but also with some kinds of interactions between the creatures and the simulated world. The implementation of such ecosystems becomes essential to encourage the development of complex adaptation strategies and to provide more complex behaviors, which were not possible to obtain in simple worlds. It will also allow regrouping evolving simulations of several species to cope with pilogeny dynamics (appearance and extinction of new species) inside one application (i.e., one world) and therefore, allow observing the relations between the evolved species. For that purpose we propose a new approach which uses classifier systems as controllers as an alternative to the use of neural networks. Several experiment were analyzed which exhibit a strong link between the fitness function and the morphology of the evolved creatures in terms of symmetry and modularity.

VII. PERSPECTIVES

The next step for our project will be to propose some paths to improve the interaction within the environment, and also to confront the evolved creatures with more complex situations to exhibit their adaptation abilities. Introducing an evolving language between the creatures could be interesting and would allow a significant increase in the number of evolving strategies. It could also be very valuable to change our morphogenesis definition allowing genotypic re-use. This could lead us to an evolving system based on the artificial embryology paradigm where a compact genetic code could lead to the growth of complex structures. Another possibility for the controller could be to compose patterns by genetic programming as we do in generic composition patterns [32].

REFERENCES

- [1] J. Bongard and C. Paul, "Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution," in *From Animals to Animats: The Sixth International Conference on the Simulation of Adaptive Behaviour*, J.-A. M. et al., Eds., 2000.
- [2] P. Eggenberger, "Evolving morphologies of simulated 3D organisms based on differential gene expression," in *Proceedings of the Fourth European Conference on Artificial Life, ECAL97*, P. Husbands and I. Harvey, Eds. MIT Press, 1997, pp. 205–213.
- [3] G. Hornby and J. B. Pollack, "Evolving l-systems to generate virtual creatures," *Computers & Graphics*, vol. 25, no. 6, pp. 1041–1048, 2001.
- [4] M. Komosinski, "The world of framsticks: Simulation, evolution, interaction," in *VW '00: Proceedings of the Second International Conference on Virtual Worlds*. London, UK: Springer-Verlag, 2000, pp. 214–224.
- [5] J. B. Pollack, G. S. Hornby, H. Lipson, and P. Funes, "Computer creativity in the automatic design of robots," *Nature*, vol. 406, 2000.
- [6] M. J. T. O'Kellym and K. Hsiao, "Evolving simulated mutually perceptive creatures for combat," in *Proc. of Artificial Life IX*, ed. I. T. Bedau M., Husbands P. and W. R. A., Eds., 2004, pp. p111–118.
- [7] S.-J. K. Yoon-Sik Shim and C.-H. Kim, "Evolving flying creatures with path following behaviour," in *Proceedings of 9th International Conference on the Simulation and Synthesis of Living Systems/ (ALIFE IX)*. Artificial Life, September 2004.
- [8] K. Sims, "Evolving 3d morphology and behavior by competition," in *Artificial Life IV*, Artificial Life. MIT Press, 1994, pp. pp. 28–39.
- [9] —, "Evolving virtual creatures," in *Siggraph '94*, 1994, pp. pp. 15–22.
- [10] T. Taylor and C. Massey, "Recent developments in the evolution of morphologies and controllers for physically simulated creatures," 2000.
- [11] J. Teo and H. A. Abbass, "Multiobjectivity and complexity in embodied cognition," *IEEE Trans. Evolutionary Computation*, vol. 9, no. 4, pp. 337–360, 2005.
- [12] T. Miconi and A. Channon, "The n-strikes-out algorithm: A steady-state algorithm for coevolution," in *the IEEE Congress on Evolutionary Computation*, 2006.
- [13] J. Ventrella, "Designing emergence in animated artificial life worlds," in *VW '98: Proceedings of the First International Conference on Virtual Worlds*. London, UK: Springer-Verlag, 1998, pp. 143–155.
- [14] —, "Attractiveness vs. efficiency (how mate preference affects location in the evolution of artificial swimming organisms)," in *ALIFE: Proceedings of the sixth international conference on Artificial life*. Cambridge, MA, USA: MIT Press, 1998, pp. 178–186.
- [15] A. Fukunaga, J. Marks, , and J. Ngo, "Automatic control of physically realistic animated figures using evolutionary programming," in *Proc. of Third Annual Conference on Evolutionary Programming*, 1994, pp. 76–83.
- [16] S. Garcia Carbajal, M. B. Moran, and F. G. Martinez, "EvolGL: life in a pond," in *Artificial Life XI Ninth International Conference on the Simulation and Synthesis of Living Systems*, J. Pollack, M. Bedau, P. Husbands, T. Ikegami, and R. A. Watson, Eds. Boston, Massachusetts: The MIT Press, 12-15 Sept. 2004, pp. 75–80.
- [17] T. Miconi and A. Channon, "A virtual creatures model for studies in artificial evolution," 2005.
- [18] —, "An improved system for artificial creatures evolution," Cambridge, MA, USA, 2006.
- [19] K. O. Stanley and R. Miikkulainen, "A taxonomy for artificial embryology," *Artif. Life*, vol. 9, no. 2, pp. 93–130, 2003.
- [20] C. Adami, C. Ofria, and T. C. Collier, "Evolution of biological complexity," *PROC.NAT.ACAD.SCI*, vol. 97, p. 4463, 2000.
- [21] A. K. Seth and G. M. Edelman, "Environment and behavior influence the complexity of evolved neural networks," *Adaptive Behavior*, vol. 12, no. 1, pp. 5–20, 2004.
- [22] A. K. Seth, "The evolution of complexity and the value of variability," in *ALIFE: Proceedings of the sixth international conference on Artificial life*. Cambridge, MA, USA: MIT Press, 1998, pp. 209–218.
- [23] J. H. Holland and J. S. Reitman, "Cognitive systems based on adaptive algorithms," *SIGART Bull.*, no. 63, pp. 49–49, 1978.
- [24] H. Olivier, S. Stephane, B. Alain, and L. Herve, "Generic classifiers system and learning behaviours in virtual worlds," in *CW '04: Proceedings of the 2004 International Conference on Cyberworlds (CW'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 113–120.
- [25] J. R. Koza, "Genetic programming," in *Encyclopedia of Computer Science and Technology*, J. G. Williams and A. Kent, Eds., vol. 39. Marcel-Dekker, 1998, pp. 29–43.
- [26] J. Klein, "Breve: a 3d environment for the simulation of decentralized systems and artificial life," in *ICAL 2003: Proceedings of the eighth international conference on Artificial life*. Cambridge, MA, USA: MIT Press, 2003, pp. 329–334.
- [27] M. M. Dalvand and M. M. Moghadam, "Stair climber smart mobile robot (msrox)," *Auton. Robots*, vol. 20, no. 1, pp. 3–14, 2006.
- [28] N. Lassabe, H. Luga, and Y. Duthen, "Evolving creatures in virtual ecosystems," in *16th International Conference on Artificial Reality and Telexistence*, to appear.
- [29] G. Abramson, "Ecological model of extinctions," p. 785, 1997.

- [30] D. Floreano and F. Mondada, "Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot," in *Proceedings of the Conference on Simulation of Adaptive Behavior*, 1994.
- [31] S. Nolfi and D. Floreano, "Synthesis of autonomous robots through evolution," 2002.
- [32] N. Lassabe, S. Sanchez, H. Luga, and Y. Duthen, "Genetically programmed strategies for chess endgame," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM Press, 2006, pp. 831–838.