Proceedings of ACM Symposium on Applied Computing (SAC 2000), pp. 404 - 407, March 2000, Como, Italy.

Solve Customers' Problems: Interactive Evolution for Tinkering with Computer Animation*

Ik Soo Lim and Daniel Thalmann
Computer Graphics Lab (LIG)
Swiss Federal Institute of Technology (EPFL)
CH-1015 Lausanne, Switzerland
iksoolim@email.com , thalmann@lig.di.epfl.ch

Abstract

This paper addresses an issue of solving *customers'* problems when applying evolutionary computation. Rather than the seemingly more impressive approach of *wow-it-all-evolved-from-nothing*, *tinkering with exisiting models* can be a more pragmatic approach in doing so. Using interactive evolution, we experimentally validate this point on setting parameters of a human walk model for computer animation while previous applications are mostly about evolving motion controllers of far simpler creatures from scratch.

Key Words: interactive evolution, genetic algorithms, synthetic actors, virtual humanoids, applications of artificial life technologies, computer animation.

1 Introduction

Artificial Life aims at understanding life-like phenomena by attempting to synthesize them from scratch. This synthesis part has also attracted interests from many application fields and some of Artificial Life techniques are employed to them: for example, evolutionary computation for computer animation. Genetic algorithms are used to synthesize stimulus-responsive motion controllers for 2D and 3D articulated figures [1][13]. The morphologies of virtual creatures in simulated 3D physical worlds and the neural systems for controlling their muscle forces are both evolved using genetic programming based on graph genotype [16]. Standard genetic programming is also applied for motion controllers for a fixed morphology [8][9][10]. Both a genetic algorithm and interactive evolution are used for generating motion controllers in a time-series representation [19][20].

However, to our knowledge, these attempts are, *in practice*, not much followed up in computer animation. Being inspired from Artificial Life, these works are done in the tradition of *it-all-evolved-from-scratch* or *it-started-without-prior-knowledge* while, given mapping functions between motion-control parameters and trajectory values, parameter tweaking is more frequently concerned in the practice of animation [11]. In this paper, we argue that *tinkering with existing models* may be a more pragmatic approach than *starting from scratch* in some applications of evolutionary techniques and we validate this point on setting parameters of a walk model in generating various walk styles for a virtual humanoid. Section 2 explains interactive evolution, a variation of genetic algorithms which is employed for the tinkering. In Section 3, we give a brief description of the human walk model and the representation of genotypes and mutation/mating operations with experiment results. Discussion and Conclusion follow.

*This work is supported in part by PAVR under the EU Training and Mobility of Researchers program.

2 Interactive Evolution

Interactive evolution provides a powerful technique for enabling human-computer collaboration. It is potentially applicable to a wide variety of search problems, provided the candidate solutions can be *produced quickly* by a computer and *evaluated quickly and easily* by a human. Since humans are often very good and fast at processing and assessing pictures, interactive evolution is particularly well suited to search problems whose candidate solutions can be visually represented [6][15][18]. While traditional genetic algorithms use an explicit analytic expression for a fitness function to be evaluated by the computer, with interactive evolution the user performs this step based on visual perception.

The beauty of interactive evolution is that the user does not have to state or even understand an explicit fitness criterion: the need is only to be able to apply it. This also frees him from tedious user specifications, design efforts, or knowledge of algorithmic details. This feature of interactive evolution is, for example, used very effectively in creating beautiful and abstract color images [15]. An initial population of images generated randomly by the computer is displayed on the screen. From the displayed set the user selects one image for mutation or two images for mating. The mating and/or mutation operations are applied to the selected images to produce a new set of progeny images, that supply the input for the next round of user selection. This process is repeated multiple times, to evolve an image of interest to the user. Evolved images may be saved and later recalled for mating with other evolved images. There are many other notable applications of interactive evolution since the inspiring work of Richard Dawkins [6] (see [3][17] for extensive reviews of it.)

3 Tinkering with a Walk Model rather than Evolving it from Scratch

Motion control of articulated figures such as humans has been a challenging task in computer animation [2]. Once an acceptable motion segment has been created, either from key-framing, motion capture or physical simulations, reuse of it is important. Much of the recent research in it has been directed towards modifying existing example motions to create a new motion [5][14][21]. In our experiment, we use a human walk model based on biomechanical data [4]. By scaling or offsetting reference motion, this walk model allows various walk styles to be generated: new motion = $s *$ reference motion $+ o$. This would be practically useful for applications such as animating crowds [12]. There being scaling and offset parameters for dozens of joint angle trajectories, manual tweaking of these parameters is a very painful process. We employ interactive evolution for computer-assisted parameter setting in generating various walk styles.

3.1 Genotype

A walk style can be described by a set of the scaling and offset parameters, s_i and o_i. In our experiment, the genotype is represented as a vector of the parameters $w = (w_1, \cdots, w_{2N}) = (s_1, \cdots, s_N, o_1, \cdots, o_N)$, where N is the number of degrees of freedom.

3.2 Mutation

Given a genotype w^{parent} representing a walk style, its mutated versions w^{child} are generated by

$$\begin{aligned} w_i^{child} &= w_i^{parent} \\ &or \\ w_i^{child} &= w_i^{parent} + d_i \end{aligned}$$

for $i = 1, \cdots, 2N$, where d_i is a displacement or perturbation factor. The choice of one or the other depends on a mutation rate indicating the probability that a given parameter will mutate during reproduction.

3.3 Mating

Mating takes two parent walk styles as inputs and uses them to produce a child walk style. The basic approach in mating is to choose a subset of the parameters from each parent and combine them to form the child. Given two genotypes of parents $w^{parent1}$ and $w^{parent2}$, their offsprings w^{child} are generated by

$$\begin{aligned} w_i^{child} &= w_i^{parent1} \\ &or \\ w_i^{child} &= w_i^{parent2} \end{aligned}$$

for $i = 1, \cdots, 2N$, where the choice of one or the other depends on a probability that a given parameter will derive from the first of the parents.

3.4 Experiments

Our experiments use, as a front end, an in-house animation software based on OpenGL. Figure 1 shows snapshots of the screen which illustrate what it looks like when evolving the walk styles. Animation of five different walk styles are shown at a time. Among them, the user selects one to mutate or two to mate; then a new generation of walk styles is generated, based on the selection, and this process is repeated. Figure 2 shows a selection of the evolved walk styles: time flows from top to bottom. It took a few minutes for each of them to evolve.

4 Discussion and Conclusion

Interactive evolution is used to generate various walk styles out of a walk engine. This kind of computer-assisted parameter tweaking can be applied to rendering and modelling process other than animation in computer graphics. Though important, *building* new renderers, modellers or motion-controllers hardly concerns many of computer graphics users in that their day-to-day work is mostly about finding input parameters that yield a desirable output under *given* renderers, modellers or motion-controllers. Manual parameter tweaking is, however, very tedious due to the multi-linearity, nonlinearity and discontinuity of the mappings between the input parameters and output values [11]. Hence, *tinkering with existing models* such as computer-assisted parameter setting of a given, say, motion-controller can be a more down-to-earth application of evolutionary computation focusing on the customers' problem than those typical attempts of *wow-it-evolved-from-nothing* for a motion controller itself.

5 Acknowledgments

We thank Paolo Baerlocher and Christophe Bordeux for their helpful discussions.

References

[1] J. Auslander, et al. Further Experience with Controller-Based Automatic Synthesis for Articulated Figures. *ACM Transactions on Graphics*, v14 n4, pp.311-336, 1995.

[2] N. Badler, C. Phillips and B. Webber. *Simulating Humans: Computer Graphics, Animation, and Control.* Oxford University Press, 1993.

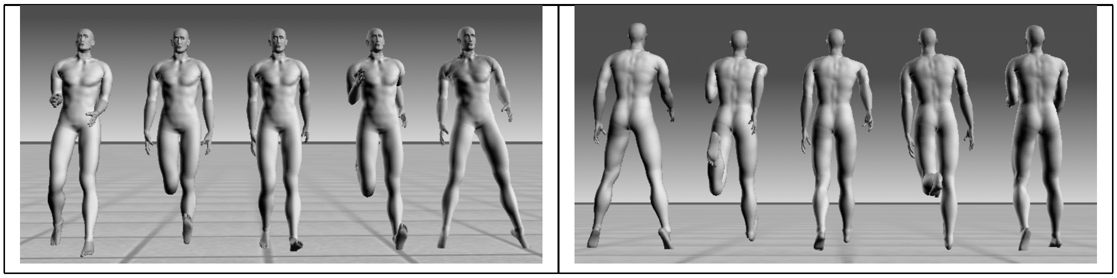

Figure 1: Typical screen looks during the interactive evolution. The user can have a view in an arbitrary direction before any selection.

[3] W. Banzhaf. Interactive Evolution. *Handbook of Evolutionary Computation.* T. Back, D. B. Fogel and Z. Michalewiez, editors. IOP Publishing Ltd and Oxford University Press, 1997.

[4] R. Boulic, D. Thalmann and N. Magnenat-Thalmann. A Global Human Walking Model with Real Time Kinematic Personification. *The Visual Computer*, v6 n6, 1990.

[5] A. Bruderlin and L. Williams. Motion Signal Processing. *Proceedings of ACM SIGGRAPH 95*, pp. 97-104, 1995.

[6] R. Dawkins. *The Blind Watchmaker*. Norton: New York, London, 1987.

[7] D. Foley, et al. *Computer Graphics : Principles and Practice, Second Edition in C*. Addison-Wesley, 1990.

[8] L. Gritz and J. K. Hahn. Genetic Programming for Articulated Figure Motion. *Journal of Visualization and Computer Animation*, v6, pp. 129-142, 1995.

[9] L. Gritz and J. K. Hahn. Genetic Programming Evolution of Controllers for 3-D Character Animation. *Proceedings of Genetic Programming 97*. pp. 139-146, 1997.

[10] Y. M. Kang, S. J. Park and E. T. Lee. An Efficient Control over Human Running Animation with Extension of Planar Hopper Model. to appear in *Journal of Visualization and Computer Animation.*

[11] J. Marks, et al. Design Galleries: a General Approach to Setting Parameters for Computer Graphics and Animation. *Proceedings of ACM SIGGRAPH 97*, pp.389-400, 1997.

[12] S. R. Musse and D. Thalmann. A Model of Human Crowd Behavior. *Proceedings of EUROGRAPHICS Workshop on Computer Animation and Simulation 97*, pp.39-51, 1997.

[13] J. T. Ngo and J. Marks. Physically Realistic Motion Synthesis in Animation. *Evolutionary Computation*, v1 n3, pp.235-268, 1993.

[14] C. Rose, M. F. Cohen and B. Bodenheimer. Verbs and Adverbs: Multidimensional Motion Interpolation. *IEEE Computer Graphics and Applications*, pp. 32-40, September/October 1998.

[15] K. Sims. Interactive Evolution of Equations for Procedural Models. *The Visual Computer*, v9, pp. 466-476, 1993.

[16] K. Sims. Evolving 3D Morphology and Behavior by Competition. *Artificial Life*, v1, pp. 353-372, 1994.

[17] H. Takagi. Interactive Evolutionary Computation - Cooperation of Computational Intelligence and Human. *Proceedings of 5th International Conference on Soft Computing*, pp.41-50, World Scientific, Iizuka, Fukuoka, Japan, October 1998.

[18] S. Todd, W. Latham and P. Hughes. Computer Sculpture Design and Animation. *Journal of Visualization and Computer Animation*, v2, pp. 98-105, 1991.

[19] J. Ventrella. Explorations in the Emergence of Morphology and Locomotion Behavior in Animated Characters. *Proceedings of Artificial Life IV*, pp. 436-441, 1994.

[20] J. Ventrella. Disney Meets Darwin: the Evolution of Funny Animated Figures. *Proceedings of Computer Animation 95*, pp. 35-43, April 1995.

[21] A. Witkin and Z. Popovic. Motion Warping. *Proceedings of ACM SIGGRAPH 95*, pp. 105-108, 1995.

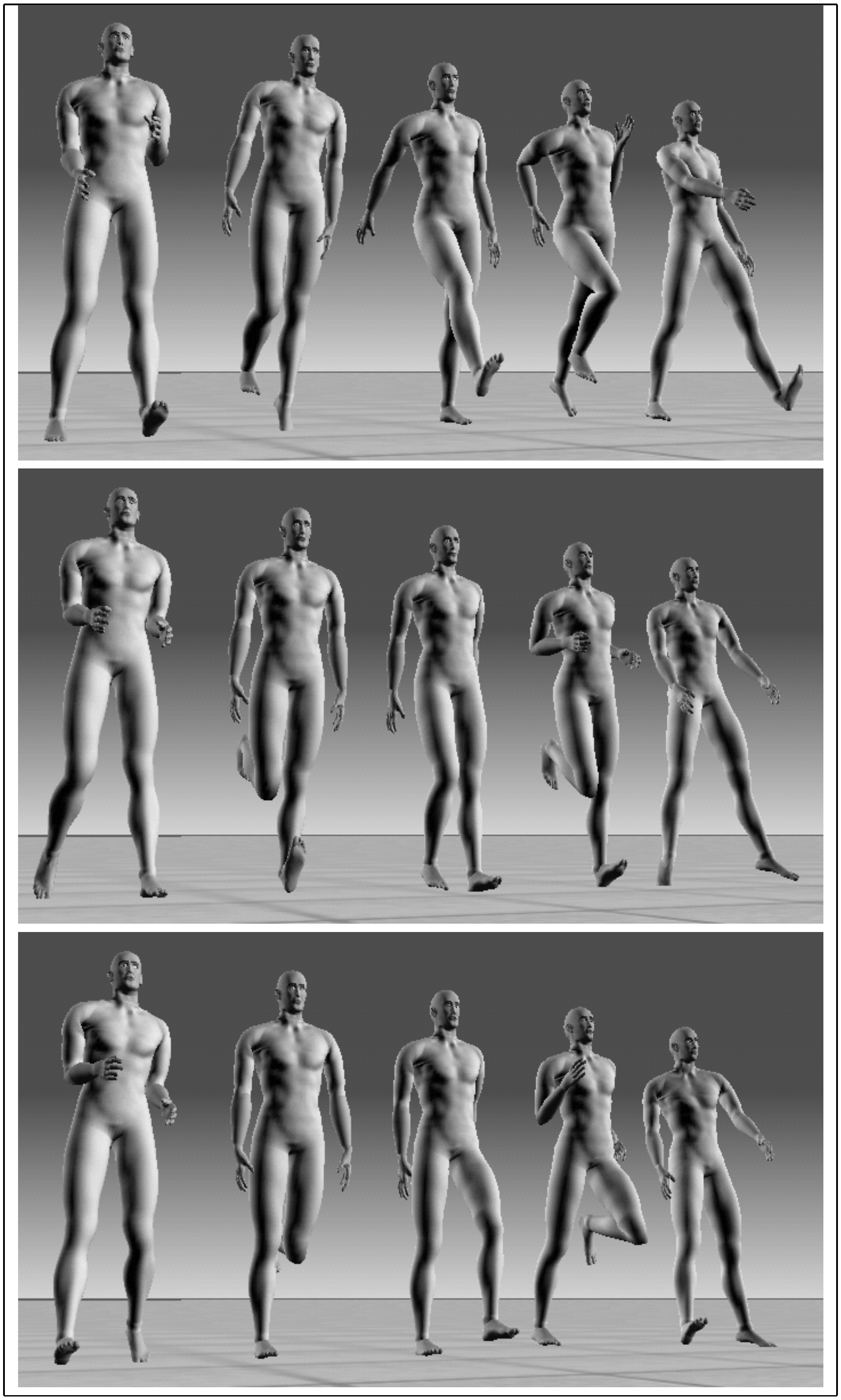

Figure 2: A selection of five walk styles among those interactively evolved. Time flows from top to bottom.