# SIMULATING VIRTUAL CHARACTER'S LEARNING BEHAVIOUR AS AN EVOLUTIONARY PROCESS USING GENETIC ALGORITHMS

**Tao Ruan Wan, *Wen Tang**

Department of Electronic Imaging and Media Communications, School of Informatics,
University of Bradford, Bradford, West Yorkshire, UK
BD7 1DP
*School of Computing and Mathematics,
University of Teesside, Middlesbrough, UK
TS1 3BA
t.wan@bradford.ac.uk , w.tang@tees.ac.uk

**ABSTRACT**

In this paper, we describe a genetic algorithm approach to simulate complex virtual character's learning behaviours as an evolutionary process. The method presented here enables virtual character to have abilities to learn for specific assigned tasks. The skill for the task can be developed and evolved through the experiences of performing the task. The animation system presented here has two tightly coupled simulation units, which are an artificial brain unit for learning and controlling and a physics-based motion simulation unit driven by simulated muscle forces.

**Keywords**: Autonomous virtual learning characters, motion control, genetic algorithms, virtual environment

## 1 INTRODUCTION AND REALITED WORK

Producing believable virtual human characters that have learning abilities in a virtual environment is an attractive topic, which would be very useful in many areas. In computer games, autonomous virtual humans that can learn for a specific task and its skill for the task can be developed and evolved would make great improvements in both fun and strategies of the game-play.

One example of this feature can be a virtual human character jumping over obstacles. By adjusting its internal muscle forces according to the level of difficulty, the virtual character is able to overcome the problem. To jump over an obstacle may be not an easy task at its first attempt, however, its ability to perform the task can be developed after a number of attempts. This kind of problem solving process can be referred as a task learning process. In real life human learning involves many complex cognitive processes. Realistically simulating virtual human characters whose behaviours reflect that of real humans demands efficient behavioural simulation algorithms, it is especially true for interactive systems such as computer games.

To develop a system, however, incorporating virtual learning characters would raise a number of challenges. From behavioural animation point of view, there are areas to be considered:

- How to design a learning behavioural control structure.
- How to store the learning information internally, evaluate and calculate feedbacks and reactions from environment efficiently
- How to couple the task learning process with characters' physical motions

Learning involves adaptation and evolution that is a feedback process in which changes made by internal subunits of the adaptive system, say human brain for this instance, mirror the external environmental changes. Many Artificial Intelligence models are applicable to simulate human learning behaviour to an extend to a certain degree of complexity [Goldberg89, Flake2000, Holland]. Among these models, we select genetic algorithms (GAs) to accomplish our goal of task orientated learning behaviour simulation for virtual characters due to the simplicity and efficiency of the algorithms. Moreover as genetic algorithms are search and optimisation procedures, the algorithms are particular suitable for task orientated simulations. Given a goal, it is possible to encode possible solutions of a simulation in a search space. The power of genetic algorithms for modelling and simulating tedious animation tasks has been demonstrated in previous research work in computer graphics community. Genetic algorithms have been used to modelling stimulus-response pairs for locomotion of 2-D stick figures [Ngo93]. van de Panne and Fiume also used genetic algorithms to optimising sensor-actuator networks for two-dimensional stick figures' locomotion [van de Panne93]. The two models described above use user-designed control structures which are fixed structures. Sims's[6] evolving virtual creatures were generated by a combination of genetic algorithms and object instancing techniques [Sims94]. The methods presented by Sims were able to generate virtual creatures with evolutionary body structures and behaviours that evolve and adopt environmental information. More recently, research work has been reported that uses neural networks for modelling artificial agent behaviour and selecting dancing styles [Crzeszczuk98, Brand2000]. The last system was trained off-line using previous collected data sets of physics-based motion actions then used to emulate a motion simulation. The work has had similar principle to the work reported in other area [Goldberg89, Flake2000].

Our system is also designed with fixed control structures. However, the differential of the system to other related work is that we extend the genetic algorithms to incorporate learning mechanisms with memory. The control structure in our system is referred as an Artificial Brain Unit (ABU) that is a feedback system with reinforcement and rewarding schemes. This approach is related to classifier system in artificial intelligence. The extension of genetic algorithms enables searching and learning to be done more effectively and efficiently compared with the conventional GA methods. Our system is constructed with emphasise on producing simulations based on the concept of learning by experience. The system presented here simulates three-dimensional virtual human characters. Given a specific task, the characters are able to develop its skills for the task through experiences of performing the task. The experience of a good activity will be stored in the memory database for improving the next attempt. This approach is, therefore, one step closer to the way that human brain and body's task learning process. Our animation system has two tightly coupled simulation units, which are a control structure for learning and a physics-based motion simulation unit driven by simulated muscle forces of the virtual human body.

It is important to note that motion control and synthesis is the second aspect concerned with the learning behavioural simulation. Motion synthesising is certainly not a new topic and many research work has been published in the past [Thalmann99, Hegron89, Girard85]. These methods very greatly in complexity and quality, most of them dealt with limited aspect of virtual human locomotion whilst others have used numerical solutions that usually have high computational cost even with today's CPU power. Our contributions to the filed of study is also the use of forward dynamics for virtual character's physical motion animation instead of the widely used inverse dynamics methods [Brogan98, So96, Boulic92]. Therefore, our system can produce realistic physically-based motion at same time still maintain good control over the motion animations.

In section two, we describe the design and construction of our control system for the learning behaviour simulation based on a human learning model. In section three, details of our implementations of genetic algorithms are presented. Implementation results and performance evaluation are shown in section four. Finally, section five concludes the research work.

## 2 THE VIRTUAL LEARNING MODEL

### 2.1 Human Adaptability in Learning

In order to simulate virtual human character's learning behaviour, a learning model has to be adapted. One of the most well known principles of human task learning behaviour is learning by experience [Jordan92]. Most of the time in our lives we learn by our direct experiences of performing a task. Learning is an intricate process that involves many aspects of cognitive activities including knowledge acquisition, observing and thinking. As each person has his/her own unique way of learning and motivations to learn, the individual's learning patterns also affects the learning process. To perform a specific task, a person's skills and abilities for the task can be developed or evolved during practices [Bransford99]. This concept can be summarised in Figure 1 and will be used to form a basis of our learning behaviour model for virtual human character's simulation. Figure 1 shows that a human learning process consists a number of key elements: background knowledge of a specific task, motivations to do the task, memory of the past experiences, individual learning pattern, and finally tries and errors through performing the task. The learning process is a process of adaptation, evolution, and decision making as a whole. Another key issue of learning is environmental feedback.

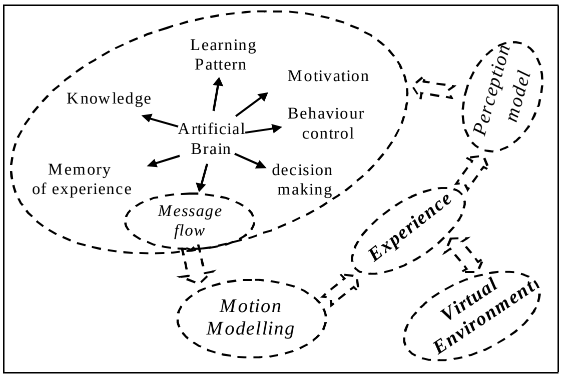


Figure 1. A General behavioural Simulation Model

Although currently it is impossible to use a computer to completely simulate all the aspects of a complex learning process, the process can be certainly approximated by some appropriate simulation methods.

### 2.2 Control Structure for Task Learning

Based on the human learning model shown in Figure 1, the simulation system, as shown in Figure 2a, is designed to consist a number of subunits. The control system here referred as an artificial brain unit (ABU) is a centre for action planning, knowledge adaptation, and task ability evolution. Assigning a specific task to a virtual human character, ABU of the character uses a set of initial motion parameters to make the first attempt to perform the task. This initial parameter set is predefined according to the basic requirements of the specific task. For example a high jump task requires a virtual athlete character to utilise its full body movements. Each individual's learning patterns and knowledge about the task are represented by predefined motion patterns that may be motion capture data. For the high jumping example, the virtual character is assumed having pre-acquisition of the knowledge of jumping by making full body movement. However, the virtual character has to improve its performance in order to achieve the target. A high jump athlete may have to make jumping attempt several times before he/she could overcome a target bar. Similarly simulating this kind of tasks requires an evolution model that approximates the evolutionary task learning process during which the ability of the virtual athlete is evolving hence improved. At each attempt in a computer simulation, the virtual athlete character has to adjust its movement by altering a set motion parameters. In our system, the motion of the virtual character is a physically-based animation by modelling a network of musical forces that acting at body joints to control the virtual character's locomotion. Learning by experience occurs during the evolution of the character's ability and its' adaptation to the environment. Evolution and adaptation are achieved through iteration and recursion of a set of motion parameters. The parameter set includes energy burst, musical forces, motion variables, etc. As genetic algorithms have been used for the evolution process, the chance to improve each next attempt is increasing that is equal to say the fitness of each attempt is increasing. Starting with an initial set of parameters and values, the control unit sends control parameters to motion unit which in turn sending actions to perform the task. Feedbacks from virtual environment about each attempt are received by the control unit and recorded as experiences. After a number of attempts, say n times, as shown in Figure 2b, genetic algorithm operations are applied that search for the next generation of motion parameters. Each set of the newly generated parameter settings is a new attempt that will be activated the character. After a number of GA generated populations, for example, m generations as shown in figure 2b, with fitness-increasing traits passed on to the offspring parameters sets, an optimal solution of achieving the target task can be found.

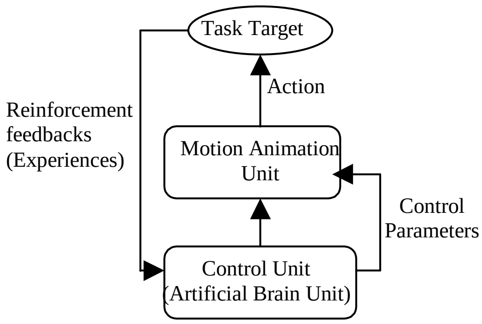


Figure 2a. Control structure of the system

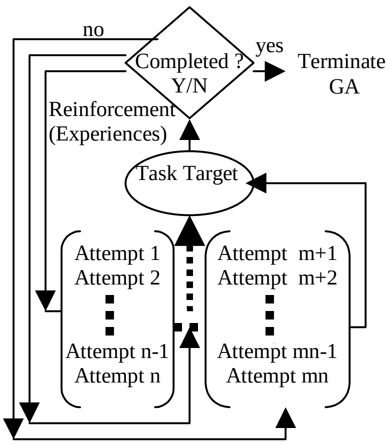


Figure 2b Genetic Algorithm operated control system

It should be mentioned that in the current system, the principles of forward dynamics has been used in the motion control unit to define the motion action of the character. In the past, using forward dynamic to animate the movements of articulated bodies has not been very successful due to the fact that the internal forces acting within the body joints were unknown over time. In order to overcome the difficulties imposed by the forward dynamics method, many researchers have used inverse dynamics as an alternative approach for simulating virtual characters' locomotion [Brogan98, So96, Boulic92]. In our system, however, forward dynamics can be used since the values of the internal forces emitted by the body muscles can be derived from the experiences simulated by the ABU.

## 3 DETIALS OF THE GENETIC ALGORITHM

A prototype system has been developed to test the algorithm. A test example was designed by assigning specific tasks to a virtual human character. For example a simple task can be the virtual human learning how to jump to a target height and develop its skill through experiences of jumping attempts.

Genetic Algorithms are search procedures based on the mechanisms of natural selection and genetic evolution. GA based methods are robust, problem independent. The massive parallelism incorporated in the algorithms enables the algorithms to have a good overall performance as a whole since each member of a population represents a hypothesis for the target solution. However, the drawback of genetic algorithms is the algorithms' convergence speed of convergence. This problem can usually resolved by optimising the size of genetic population with respect to the number of variables encoded in each member of the population.

The first concern of implementing a GA based algorithm is to derive a state variable space. In our case is to derive a state variable space from the physics-based model considered. This means to encode the potential solution to the task problem into a form that is suitable for a GA based algorithm to operate. In a GA implementation, each candidate solution is carefully defined in the state variable space and regarded as an individual or a 'child'. A collection of all current individuals forms a population. The individuals with fitness rates together with some individuals selected randomly from the current population will be responsible for producing the next generation of a population. If the specific task for the virtual character in our system is to learn how to jump to reach a target height, there is need for us to construct a character body model.

For the jumping task or similar type of tasks, the physical motion of the character is driven by a set of motion parameters and values, which are concerned with the full body movement. This movement can be considered as a number of phases, in particularly fast running in a very short distance and jumping. These movements of our virtual character are driven by a 'low level', parameter space that are formed by the characteristics of muscle forces acting at the body joints. In contrast with the 'high level' parameters, such as speed of walking, step length and frequency etc., which can be obtained directly through motion capture data [Li2000, Cavaza98]. The designed parameter space is suitable for physically-based motion animation and genetic operations. Figure 3 illustrates a knee joint illustrates utilising two muscle pairs to fully control its movement in three dimension.

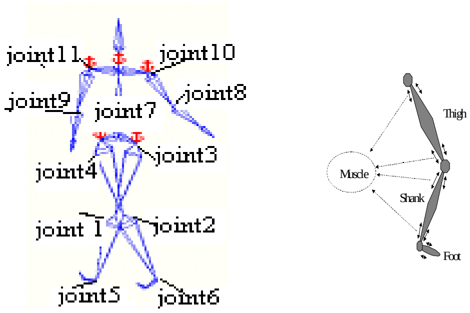


Figure 3: An articulated character body with muscle pairs

The initial pattern of muscle forces for a particular task is derived from the experience memory database that may be only one type of time-domain sequence or a combination of several types of time-domain sequences depending on how complex the task is. It is also important to keep the parameter space small since the smaller number of parameters for searching operation, the higher efficiency the algorithm can perform. For the jumping task, we have designed a set of parameters to act on the muscle units at each of the body joints. There are total 11 joints in our model as shown in Figure 3. However, for simplicity, for the jumping simulation, genetic operations are only applied to the lower body joints while as the up body movements are user defined.

Each muscle unit are considered as energy emitters. A set of parameters related to the body muscles are designed, which are values of muscle forces, energy limits, energy burst types, muscle state, action period, and kinematics constrains that is designed to guarantee the synthesised locomotion is physically plausible. Each set of the state variables that are needed for a motion movement are stored in a string data type. Some of the elements in each set have a

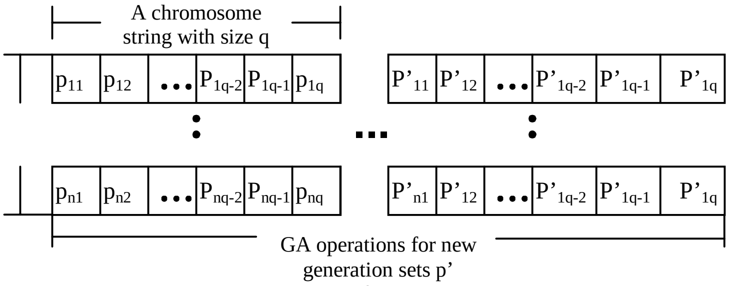


Figure 4. Chromosome representation of the parameter space as an optimisation problem

predefined value range respectively. The state variable space is represented in a form of a set of parameters or state attributes. For a typical body joint i, the parameter sets for the joint are shown in table 1.

| string element | parameter | value range | parameter type at joint I |
|---|---|---|---|
| $P_{i1}$ | $f_i$ | $[f_{min}, f_{max.}]$ | muscle force |
| $P_{i2}$ | $p_i$ | $[0.0, p_{target}]$ | position |
| $P_{i3}$ | $v_i$ | $[v_{min}, v_{max.}]$ | velocity |
| $P_{i4}$ | $a_i$ | $[a_{min}, a_{max.}]$ | acceleration |
| $P_{i5}$ | $E_i$ | $[E_{min}, E_{max.}]$ | energy burst |
| $P_{i6}$ | $C_i$ | $[C_{min}, C_{max.}]$ | kinematics constrain |
| $P_{i7}$ | $t_i$ | $[t_{min}, t_{max.}]$ | action period |
| $P_{i9}$ | $ES_i$ | Boolean | muscle state |
| $P_{i10}$ | i | [1 ~7] | joint index |

Table 1: List of the state attributes

In the implementation, as shown in Figure 4, those parameters are represented as real numbers and constructed in a form of chromosome string form, which are therefore suitable for a GA based method to manipulate.

Conventional Genetic Algorithms usually need a searching space with a very large population, which may be over hundreds or thousands of individuals and the initial population is created randomly. This is not a case suitable for simulating character's learning behaviour. In our system, the virtual character starts its initial attempt by searching a much small space initialised by a set of well-planed or experienced individuals or seeds. We have selected a population size of ten individuals. Such small size is compromised to the efficiency of searching operation since our goal is also producing real-time learning behaviours. Therefore initial planning, memory of the individual pattern, and the past experience of the task are important in this case. After the virtual character has tried all the potential individuals in a populations, its performance will be analysed and those individuals with good performance record, i.e. meet a fitness criteria, will be chosen together with some randomly generated seeds to be responsible for creating a new generation of possible individuals. Award will also be received by the ABU for a good attempt and distributed over the entire population. Equations 1-3 are used to calculate the fitness of each individual.

Equation 1 calculates the raw fitness of each individual in the population. Raw fitness is the height made by each attempt divided by the target height. If one attempt only has a little improvement compared with another attempt, there would be a very small distinguish between the two fitness values. This would be hard for the searching to act on. To over come this problem, a scaled fitness calculated by equation 2 is used so that the improved attempt is twice as fit as the other attempt. Equation 3 is used to calculate normalized fitness by summing up all of the scaled fitness scores for every member in the population and dividing by the scaled fitness of each individual of the population. In this way, the selections made a genetic algorithm are regardless of the magnitude of the raw and scaled fitness.

$$f^{raw} = \frac{attempted\ height}{target\ height} \qquad (1)$$

$$f^{scale} = 2^{f^{raw}} \qquad (2)$$

$$f_i^{norm} = \frac{f^{raw}}{\sum_{j=i}^{n} f_j^{scale}} \qquad (3)$$

Where n is the total number of individuals in a population. Decisions that have been made on which individual should be selected to be a parent seed for next generation are based on random selections with probability equal to its fitness. Two selected seeds are paired up hence crossover and mutation operations can be applied. A relatively high crossover rate enables the healthy genes from both parents can be shared by their offspring. Compared with crossover rate, a small mutation rate introduces genetic diversity. The newly generated offspring from the parents is a set of parameters to be sent to the motion animation unit for new action.

The search process is completed once the character has reached the target. In this way, the required searching space is reduced and the efficiency for reaching an optimised solution can be increased significantly.

## 4. EXPERIMENTAL

The resultant performance of a virtual character for the task can be assessed by a set of rules or directly by the goal that the character intended to achieve, in this experiment is a target height. Once the character performed an action, the resulting performance will be fed back to the ABU for comparison and analysis. Over a period of time, as the improvements of performance through the experience, fitness-increasing traits, which are the parameter settings in our case, have being passed over generations of attempts, the virtual human in our system is being trained to do the task better. Although there may be some temporarily performance dropping, the ability of the character for the task is evolving and improving

| | Generation 1 | Performance |
|---|---|---|
| C1 | 2555000505000000 | 33.09 |
| C2 | 2555010305000201 | 33.13 |
| C3 | 2555020405010001 | 33.18 |
| C4 | 3159649581096811 | 40.02 |
| C5 | 2764421143160468 | 45.50 |
| C6 | 3060068945235030 | 31.39 |
| C7 | 2855975017310557 | 47.72 |
| C8 | 2660833884013314 | 26.55 |
| C9 | 3259674554198852 | 43.58 |

| | Generation 2 | Performance |
|---|---|---|
| C1 | 3056975017310557 | 51.30 |
| C2 | 2659421143160468 | 27.15 |
| C3 | 2960674554198892 | 31.73 |
| C4 | 2659421143160366 | 27.13 |
| C5 | 3056975017310556 | 51.31 |
| C6 | 2756612813926758 | 38.26 |
| C7 | 2960674555198692 | 31.71 |
| C8 | 2560390754322098 | 23.58 |
| C9 | 2555000000000000 | 33.09 |

Figure 5: Example of the decimal string representation of genes

Figure 5 shows an example of the decimal string representation of genes, which is a part of two generations in a jumping optimisation process. It can be seen from this figure that the general performance of jumping is not significantly improved in the second generation, however, there is one with exceptional good performance. It will be proved that the general performance will be improved greatly as more generations are being produced.

Figure 6 shows the result of the performance evaluation. A population of a set of the parameters was initialised in the first jump trial. In this experiment, each trial consists of nine jump attempts, which are the population of the current generation. The virtual character starts to make a number of jumps by utilising its muscles and at the same time keeping itself in a dynamic balance. The performance of by the mean value of the top 6 jumps. As shown in this figure, the convergence towards the goal is evident and stable and it reaches an optimisation point after a total of approximately.90 attempts.

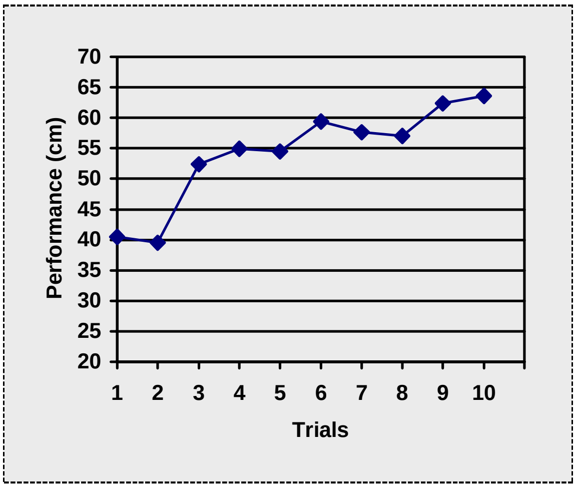


Figure 6: Average height values reached by each generation

Figure 7 shows an animation sequence of a jumping task. As shown in this figure, this is not a simple task but a combination of running and jumping. In order to complete a jumping action, the articulated human figure has to make a number of complex manoeuvres by using its muscles and at the same time keeping a good dynamic balance.

Figure 8 shows a simulation sequence of jumping in a virtual environment. The virtual character was trying to collect apples from an apple tree but it may be not an easy task for him. He might not be able to reach the target at his first jump attempt. In order to reach his goal, he had to jump as high as he could by learning through a number of attempts. Let us suppose that he can complete the task physically, it is only a matter of skill development. He has to learn by his experience of each attempt until his skill is fully developed to reach the apple. Our system is able to simulate this process in real-time.

## 5 CONCLUSION AND FURTURE WORK

In this paper we presented a novel approach in simulating virtual human's learning behaviours for interactive virtual environment applications. Our contribution to the field is that we have proposed a concept of article brain unit (ABU) that is a control unit of the character's virtual brain. The ABU model for task learning behaviours is based on a human learning model. This is not to say that ABU is a true simulation of real human brain activities rather a simulation component that models of a number of aspects of human brain learning activities. This ABU model can be also extended to model different types of learning behaviours. We have developed simulation system based on this idea for virtual character's learning behaviour simulation. The result produced by our system is encouraging. However, complete satisfactory behavioural simulations may involve research from many disciplines ranging from computer science, mechanics, Physics and Biology etc. Although the current work is still at early stage, a framework for simulating complex virtual human learning behaviour has been developed in our research. The learning model also needs the behaviour rules but it is one step further to the previous rule-based behavioural animations.

Our work is to study not only motion behaviours but also concerned with the learning how to conduct these motions through experiences. The concept of artificial brain presented in this work provides a method for autonomous virtual humans to evolve its task performance through experiences in a virtual world. It also thanks to the muscle-driven model developed here that makes it possible to implement the research idea as a whole. The future work will be continuing to improve the simulation system in order to simulate more complex human learning behaviours for example learning dance, and to look at applications in computer games and virtual training systems. Many artificial intelligent algorithms can be incorporated with the genetic algorithm to simulate interesting behaviours for virtual characters. The challenges are admitablely concerned with the efficiency, realism, and control of the simulation. The future direction of this work will be also to make extensions to the current system by addressing these challenges.

## REFERENCES


[Boulic92] Boulic R. Thalmann D. *Combined Direct and Inverse Kinematic Control for Articulated Figure Motion* Editing, Computer Graphics Forum 1992; 11(4), p.189-202.

[Brand2000] Brand, M., Hertzmann, A., *StyleMachines*, Computer Graphics *SIGGRAPH 2000 Proceedings*, July 2000, pp. 183-192.

[Bransford99] Bransford J.D., Brown A.L. and Cocking R.R., *Brain, Mind, Experience, and School*, National Academy Press, Washington, D.C. 1999.

[Brogan98] Brogan D.C., Metoyer M.P, Hodgins J.K., *Dynamically Simulated Humans in Virtual Environments*, IEEE Computer Graphics and Applications, 1998, 18(5), 25-34.

[Cavaza98] Cavaza M. Earnshaw R, Magnenat-Thalmann N. and Thalmann D., *Motion Control of Virtual Humans*, IEEE Computer Graphics and Applications , 1998, 18 (5), p 24-31.

[Crzeszczuk98] Crzeszczuk, R, Terzopoulos, D., Hinton, G, *NeoroAnimator: Fast Neoral Network Emulation and control of Physics-Based Models*, Computer Graphics (SIGGRAPH 98 Proceedings), July 1998, pp. 183-192.

[Flake2000] Flake, G.W., *The Computational Beauty of Nature.*Cambridge, Mass.: The MIT Press, 2000

[Girard85] Girard M. Maceijewki A.A. *Computational Modelling for the Computer Animation of Legged Figures*, SIGGRAPH'85, computer Graphics 1985, 19(3), p263-70.

[Goldberg89] Goldberg, D.E.,*Genetic Algorithms in search, optimisation, and machine Learning*. Reading, Mass.: Addision-Wesley, 1989

[Hegron89] Hegron G, Palamidese P., Thalmann D., *Motion Control in Animation, Simulation and Visualization*, Computer Graphics Forum 1989, 347-52

[Holland ] Holland, J.H. *Adaptation*. In R. Rosen & F.M. Snell (Eds.), *Progress in theoretical biology IV* pp.263-293, New York: Academic Press.

[Jordan92] Jordan, M.I., Rumelhart, D.E., *Supervised Learning with a distal Teacher*, Cognitive Science, 16, 1992, p307-354.

[Li2000] Li, L. X. Liu, X., *Simulating Human Walking on Special Terrain: up and down Slopes*, Computer & Graphics, 24, 2000, p453-463.

[Ngo93] Ngo, J.T., and Marks, J., *Spacetime Constraints Revisited*, Computer Graphics, Annual Conference Series, 1993, pp.335-342

[Sims94] Sims, K., *Evolving Virtual Creatures*, Computer Graphics, Annual Conference Series, 1994, pp.15-22

[So96] So, H. Badler, N. I *Animating Human Locomotion with Inverse Dynamics*, IEEE Computer Graphics and Applications, March 1996, p50-59.

[Thalmann99] Thalmann, D., Farenc, N., Boulic, R., *VirtualHuman Life Simulation and Database: Why and How*, Proceedings International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), IEEE CS Press.

[van de Panne93] van de Panne, M., and Fiume, E., *Sensor-Actuator Networks*, Computer Graphics Annual Conference Series, 1993, pp.335-342

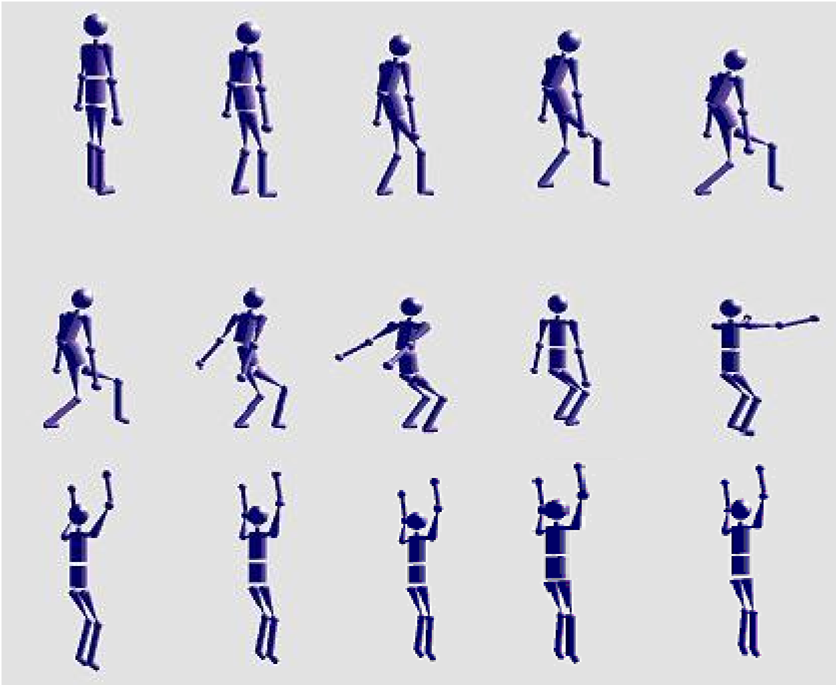

Figure 7. A simulation sequence of an articulated virtual human character jumping

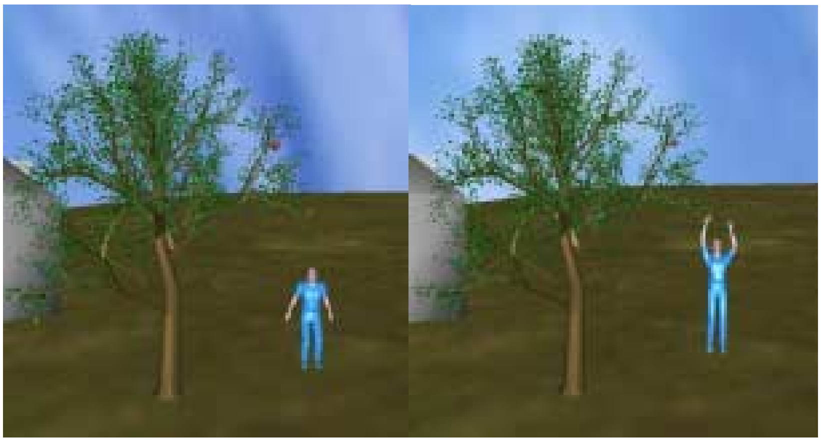

Figure 8. Four instances of a virtual human jumping in a virtual world