The Visual Computer

Motion and behaviour modelling: state of art and new trends

Eva Cerezo[1], Alfredo Pina[2], Francisco J. Serón[1]

[1] Advanced Computer Graphics Group (GIGA),
Computer Science Department,
Technical School of Engineering,
University of Zaragoza, Zaragoza, Spain
E-mail: ecb@phoebe.cps.unizar.es
seron@posta.unizar.es
[2] Mathematics and Computer Science Department,
Public University of Navarra, Pamplona, Spain
E-mail: pina@upna.es

The main task of a computer animator is to make the objects of a purely synthetic 3D world move realistically. The great number of computer animation techniques that already exist and the new ones that are continuously appearing, are the result of a multidisciplinary exchange of ideas. Our main goal has been to provide a classification of computer animation systems focused on motion and behaviour modelling from the point of view of control techniques. We also present the historical evolution in these areas, providing an extensive bibliography and highlighting some historical milestones. Finally, a set of Internet adresses of the most relevant research groups in this area present on the Web is also given.

Key words: Computer animation – Motion modelling – Behaviour modelling – Control techniques – Synthetic actors

Correspondence to: Francisco J. Serón

1 Introduction: the historical evolution of computer animation

Computer animation covers a wide range of techniques that apparently have very little in common. Computer animation systems work with one or more elements that vary in time; several parameters have to be controlled within every element. It is not an easy task to define the best way to carry out such control.

Computer animation techniques have been moving away from traditional animation techniques and have come closer to those of mechanics, robotics, artificial intelligence and so on. Table 1 shows some historical milestones in the evolution of computer animation. The first and rudimentary systems were developed by the end of the 1960s and only helped to automate some traditional animation tasks. During the 1970s, computer animation began to be applied to task modelling, in-betweening calculation and postproduction stages. During the 1980s, script-based systems, as well as kinematic movement generation, were first started. The search for realism present in other computer graphics disciplines is also reflected in computer animation, where physical laws (i.e. dynamics) made it possible to obtain natural movement in the late 1980s. The progressive preference of the term *simulation* to the term *animation* also shows this tendency. It is important, in any case, to distinguish between animation and simulation (Hegron et al. 1989). In the first case, the objective is to generate, in different ways, frames that are sequentially visualized at a given rate to reproduce a sequence. Simulation can be defined as an animation sequence that reproduces natural phenomena in our surroundings. Simulations are based on mathematical and physical models so that the user can visualize their temporal evolution. To work in real time, an image has to be generated in less than 1/15 s. This very fact limits the amount of calculus allowed for generating each frame. It is possible to generate the animation in another way, not in real time, but by generating the frames one by one and storing them. Later, they are projected at a rate of 25 or 30 frames/s. The time spent in generating one frame can vary from a couple of seconds to several hours, depending on the complexity of the scene and the equipment available. The start and extension of so-called *physically based modelling* has had two important consequences for computer animation:

The Visual Computer (1999) 15:124–146

Table 1. Some historical milestones in the evolution of computer animation

Before 1970	1970–1985	1985–1987	1988–1990	1990–1995	after 1995
Traditional computer animation starts to use computer techniques to optimize current methods BEFLIX. (1964) and EXPLOR. (1970) by Knowlton	Animation systems use a keyframe image base for motion control MSGEN by Burtnyck and Wein (1971)	Film Tony de Peltrie with facial expressions and speech by Lachapelle et al. (1985)	Integration of kinematics and dynamics in simulations, by Isaacs and Cohen (1988)	Use of artificial life starts in motion control by Beer (1990)	Networked environments with distributed users VLNET by Noser and colleagues (1996) Real-time animation of realistic computer actors CyberTennis (Telecom Interactive'97) CyberDance (Computer Animation'97) by Magnenat-Thalmann and colleagues
First rudimentary animation systems appear GENESYS by Baecker (1969)	First 3D human facial animation by Parke (1972)	Beginning of animation of legged figures by Girard and Maciejewski (1985), Wilhelms and Barsky (1985), Armstrong and Green (1985)	SpaceTime constraints introduced by Witkin and Kass (1988)	1991 Virtual reality emerges: ACM SIGGRAPH Convention Galeria of VR Applications Tomorrows Realities (Chair Steve E. Tice)	
First computer-animated film: "Two-gyro gravity-gradient attitude control system" by Zajac (1963)	First Physically based muscle-controlled face model by Platt and Badler (1981)	Parametric keyframe-based animation by Kochanek and Bartels (1984), Steketee and Badler (1985)	Artificial life was born by Langton (1989)	Animating behaviour by Maes (1990) and Brooks (1991) van de Panne and Fiume (1993)	Real time integration of Virtual and real elements for TV ESPRIT VISTA project by several European partners
	First computer human animation by Badler (1982) and Calvert (1982)	Introduction of free-form deformations by Sederberg and Parry (1986)	New concepts: Simulation vs. Animation, by Hegron et al. (1989)	Genetic programming for motion control by Ngo and Marks (1993) and Sims (1994)	Interaction with virtual actors in artificial environments Jack by Badler et al. (1997) Olympic bicycle race by Brogan et collegues (1998)
	Concepts of Motor control, task-level animation and motion planning by Zeltzer (1982)	First physical models for cloth animation by Terzopoulos and collegues (1987)	New concepts: Physically based Modelling by Barr (1987)	Action-perception schema vs. script-based animation by Tu and Terzopoulos (1994)	Computer actors who learn, improvise, reason … . Virtual Theatre by Hayes Roth et al. (1997) IMPROV by Goldberg et al. (1997)
	Use of systems of particles in the film Star Trek-II by Reeves, Lucas Film (1982)	Concept of behaviour in computer animation by Reynolds (1987)		Natural language use, speech-based control AnimNL by Geib et al. (1994) & Sodajack platform by Webber et al. (1994)	
	First animation languages appear: ASAS by Reynolds (1982)	Concept of synthetic actor by Thalmann et al. (1987)		Designing animal and human behaviour by Blumberg (1995)	
	Global deformations proposed by Barr (1984)	First important human computer animated film: "Rendez-vous à Montréal", by MIRALab (1987)			

- As it uses complex mathematical models to control the movement of bodies and other parameters, computational requirements are important, and in general, very difficult to achieve in real time.
- Motion control becomes complicated and not very intuitive.

The next section shows how several research groups have made a big effort to lighten these problems.

The introduction of the concept of the *synthetic actor* in the late 1980s brought about a new revolution in computer animation. Since then, synthetic actors have been used intensively, not only in animation systems, but also in military systems, ergonomics, learning modules, teleoperation, entertainment and so on. These actors should be able to move in as realistic a way as possible, according to physical laws, but should also be capable of tasks development and decision making. The application of techniques stemming from robotics and artificial intelligence becomes clear now: the animated actor will be a simulated robot in a simulated world. The shift from purely reactive behaviour to another type of behaviour in which the individual acts according to his/her experience (individuals capable of learning) is a step forward in behaviour modelling.

The introduction of virtual reality in the 1990s and its progressive expansion and popularity, have opened a new world of possibilities for synthetic actors and for the simulation of physical phenomena. Distributed complex 3D virtual environments, capable of real-time interactions, represent a new challenge in the development of semiautonomous or independent actors.

New and powerful computers and the use of parallelization techniques have made it possible to handle the increasing computational complexity that current animation systems require.

This of state-of-the-art report focuses on two specific parts of computer animation systems: motion and behaviour modelling. A classification of current computer animation systems is proposed in the next section. Sections 3 and 4 deal with the development of tools and techniques used in motion and behaviour modelling, respectively. An appendix with Internet references of the most representative and innovative research groups on these areas is also provided.

2 Classification of computer animation systems

One way to classify the large number of animation techniques that already exist and the new ones that are continuously appearing is to focus on the control systems they use. Motion control represents the heart of any animation/simulation system. It determines the interface friendliness and the kind of motion and deformation that can be considered, as well as its range of applicability.

High-level animation systems allow users to specify movement in an abstract way, whereas low-level systems require the user to specify all the motion parameters. Traditionally, one can distinguish three families of control:

- *Descriptive or guided systems*, which reproduce an effect without knowing about the causes
- *Generative models*, which offer a causal description of objects motion
- *Behaviour or task-oriented models*, which simulate autonomous beings with perception, decision, action and communication abilities.

In Table 2, we present a classification of computer animation techniques based on control models. This is the classification we follow and detail in the rest of this report.

3 Evolution of motion modelling in animation systems

In this section we analyse the first row of Table 2, which presents the evolution of motion synthesis techniques. First, we present descriptive models; later, we introduce physically based modelling systems; and finally, we discuss task-oriented modelling.

3.1 Descriptive or guided models

In these models, each object is described by a number of parameters. The model explicitly describes their variations in time. We can include a number of systems in this group, which we now describe.

Table 2. Animation techniques classification

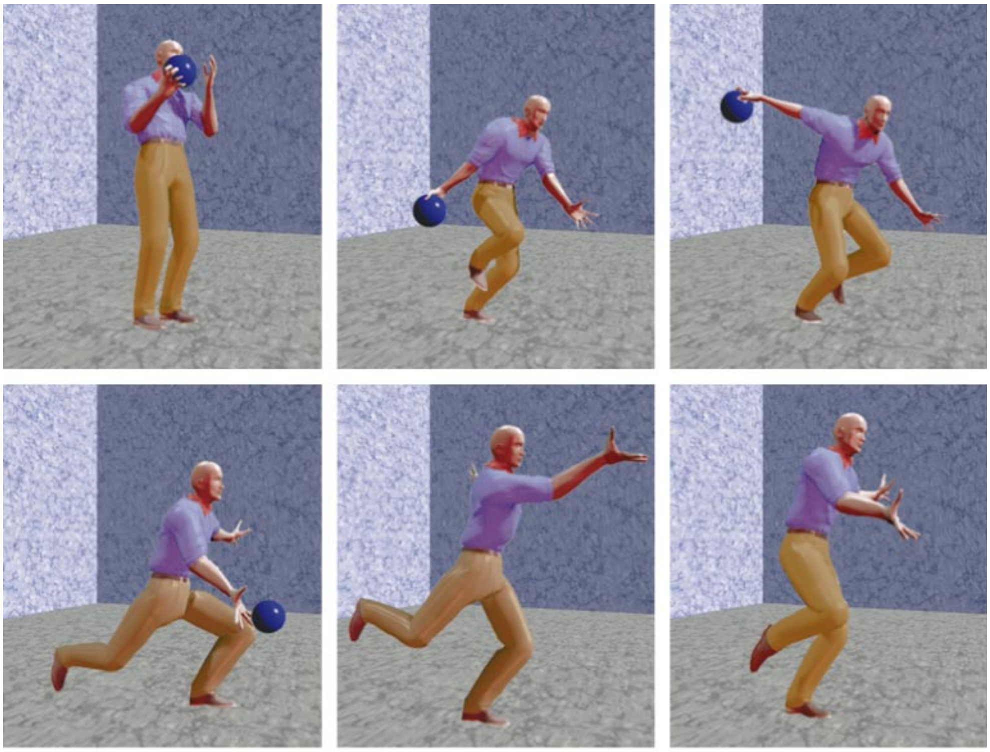
	Control Models	Animation systems
Motion Modelling	Descriptive or guided models	Keyframe systems Procedural systems Script systems Direct and inverse kinematic systems Systems with geometrical deformations Motion capture and/or rotoscopy systems
	Generative models Physically based systems	Dynamic systems Direct Inverse Dynamic systems solved By integration With preprocessed control With matrix control By nonlinear optimization Simplified Dynamic systems Recursive formulations
	Task-oriented modelling	Motor control/low level Genetic algorithms Sensor/actuator networks Finite-state machines
Behaviour modelling	Behaviour-based or task-oriented modelling	Motor planning/high level Distributed systems Local interactions Stimulus reponse Perception and action Automata systems Algorithmic systems Software agents and artificial life Rule-based systems Fuzzy logic systems Genetic algorithms Expert systems Neural networks Action selection

3.1.1 In-betweening

Animation systems can be based on *keyframing* in which the animator specifies the system's kinematics by means of giving the parameter values in the keyframes. In-betweenings are calculated by having the computer apply an interpolation law that can be linear, constrained to a mobile point (Magnenat-Thalmann and Thalmann 1990), based on splines (Kochanek and Bartels 1984; Bartels et al. 1987; Spencer-Smith and Wyvill 1989) or on quaternions (Shoemake 1985, Barr et al. 1992, Jüttler 1994). In these schemes, control over the animation is total. Nevertheless, if the number of parameters is considerable, motion specification becomes a tedious task (for instance, if we want to animate a human, even with a very simplified model with 22 links, more than 70 parameters – angles and references – have to be specified in each frame). Furthermore, the motion obtained can move away from a real one. Most commercial animation packages make extensive use of in-betweening: Fig. 1 shows several keyframes of a short animation created with Fractal Design Corporation's Poser.

3.1.2 Procedural models

A *Script-based systems* were the first control models to be developed. The animator has to write the script using an animation language. The best well-known systems are:



1

2

Fig. 1. A set of keyframes obtained with Fractal Design Corporation's Poser 2

Fig. 2. Visualization of the set of kinematic chains used to control the position of a tennis player. Note the Inverse Kinematics (IK) manipulation gizmo used to position the effectors

- ASAS (Reynolds 1982) is the first system that introduced the concept of "actor" (an object with its own animation), with communication and synchronization among actors. It posseses it own language, which is an extension of LISP.
- MIRA (Magnenat-Thalmann and Thalmann 1985) uses its own language (CINEMIRA) which is an extension of PASCAL. The system also has an editor (the ANIMEDIT), to avoid the animator having to program to write the script.

B Systems that use *direct* and *inverse kinematics* to develop algorithms for generating complex motion, such as human walking and various animal movements form the second type of procedural model. These systems are based on biomechanical and biological studies. Instead of specifying positions, the animator specifies parameters that condense the essence of the motion and allow its individualisation. In inverse kinematics, if the position of the final effector (open chains) is given, the calculation of the intermediate links is automatic (Paul 1981). Figure 2 shows an articulated figure positioned with inverse kinematics using the ARTIST package[1]. The kind of equations these systems work with is highly nonlinear; therefore, specific techniques have to be applied. Two groups of methods to solve the problem can be found in inverse kinematics systems:

1. The jacobian methods
2. Nonlinear programming techniques

As examples of the first group, we should mention:

- Girard and Maciejewski (1985) develop a general model for legged locomotion in their PODA system.
- Boulic et al. (1990) also propose a human walking model. Later, they propose a combined inverse/direct method that allowes them to use prerecorded motion as reference motion and to add user motion restrictions as a secondary task to be accomplished (Boulic and Thalmann 1992; Boulic et al. 1994).
- Mas-Sansó and Thalmann (1994) develop an algorithm for automatic grasping for a system that works with synthetic actors. It is based on a grasping taxonomy and both direct and inverse kinematics is used.
- Bruderlin et al. (1994) characterize human walking both with locomotion parameters (forward velocity, step length and step frequency) and 15 attributes that personalize motion. This motion generator is the one used in the general-purpose system for human animation "Life Forms" (Calvert et al. 1993). They also consider the grasping problem, where inverse kinematics are very useful to position shoulder, elbow and wrist once the hand is placed.
- Boulic et al. (1995) and Mas-Sansó et al. (1996) introduce the term *inverse kinetics* to express the combination of joint kinematics and mass distribution.

As an example of the application of nonlinear programming, we can mention the Jack system for animation of human figures (Fig. 3) developed at the University of Pennsylvania. The nonlinear equations are solved by means of a potential function that measures the distance to the target position. Nonlinear programming techniques help them to minimize the function (Zhao and Badler 1994).

C A third group of procedural models is formed by systems that deal with deformations geometrically. See Barr (1984), Sederberg and Parry (1986), Coquillart (1990), Coquillard and Jancène (1991), and Hsu et al. (1992) for information on free form deformations, see Preston and Hewitt (1994) on NURBS, and Magnenat-Thalmann et al. (1988) on joint-dependent local deformations.

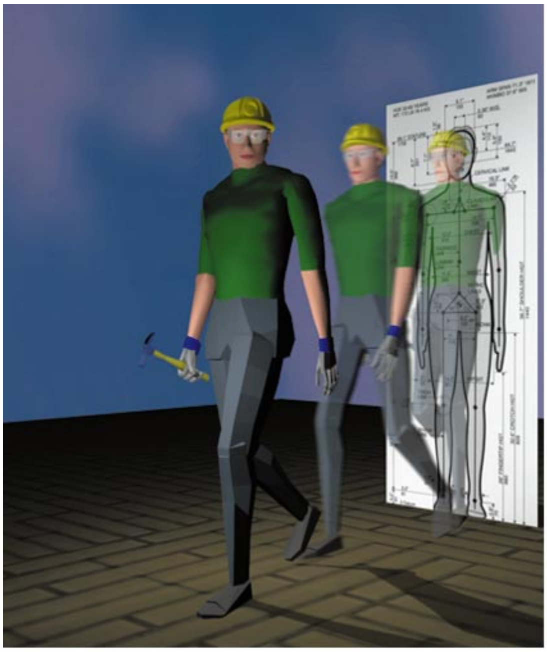
3.1.3 Motion capture and rotoscopy

Techniques have been based on live *motion capture* (Meyer et al. 1992; Bodenheimer et al. 1998) (Fig. 4) and *rotoscopy* (Luo et al. 1992) from films or video. These techniques are now being extensively used for body motions as well as for facial expressions and speech, and have given rise to the development of new editing methods to treat recorded motion. They reuse and adapt captured motion and use various algorithms such as:

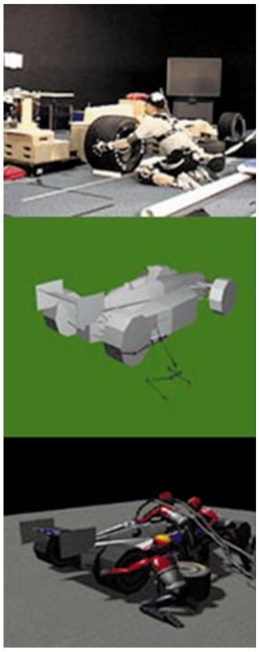
- Multiresolution filtering to personalize motion, multitarget interpolation and smoothing techniques for motion concatenation and blending

---

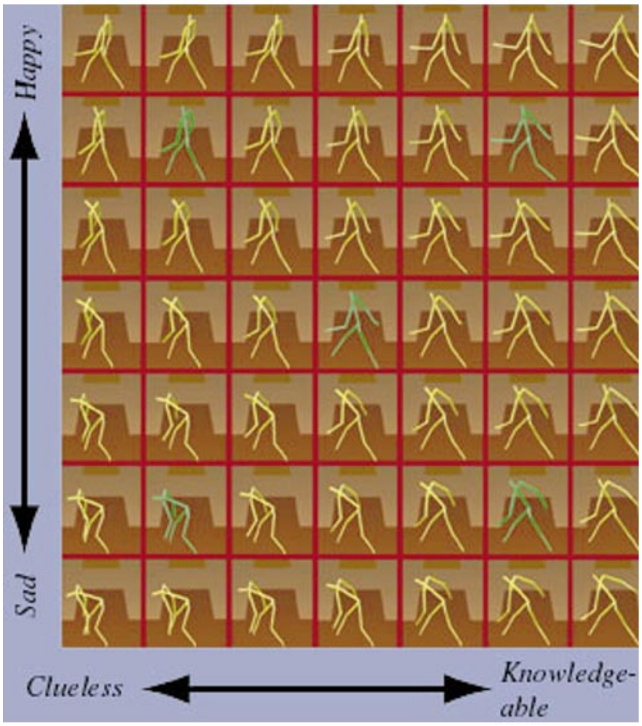
[1] Animation Package for Real-Time Simulation (ARTIST), ESPRIT Project E20102. Developed by LISITT (University of Valencia, Spain), GIGA (University of Zaragoza, Spain), Norks Regnesentral (Norway), APD (Spain) and Art & Magic (Belgium).



3



4



5

Fig. 3. Jack character. Image courtesy of Engineering Animation, Inc

Fig. 4. Example of motion-capture data transferred to 3D animation (Bodenheimer et al. 1998)

Fig. 5. Walking parametrization along two emotional axes: happiness (vertical) and knowledge (horizontal). Verbs and adverbes: multidimensional motion interpolation (Rose et al.1998)

(Bruderlin and Williams 1995; Rose et al 1998) (Fig. 5)
- Displacement functions (Snibbe 1995)
- Fourier developments on experimental data to add emotions to previously captured human motion (Unuma et al. 1995)
- Motion warping (Witkin and Popovic 1995)
- Contour tracking in multiple images (Hoch and Litwinowicz 1996) and facial expression analysis for virtual conferencing (Eisert and Girod, 1998)

3.2 Generative models: physically based modelling

3.2.1 Direct and inverse dynamics

It was the search for realism that produced the development of new control techniques. In particular, physically based modelling (Barzel 1992) has made it possible to consider natural phenomena applying simulation techniques. Mechanical simulation works with three kinds of objects:

- Articulated rigid objects (already mentioned)
- Deformable objects (Terzopoulos et al. 1987): with the appearance of systems that work with synthetic actors, the interest in human body deformations (Gourret et al. 1989; Boulic et al. 1995; Volino et al. 1996), including facial ones (Waters 1988; Platt and Badler 1981), has increased. Many groups are also working on cloth deformations (Fig. 6) (Kunii and Gotoda 1990; Carignan et al. 1992; Deussen et al. 1995; Louchet et al. 1995; Ng et al. 1995; Volino and Magnenat-Thalmann 1995; Ling et al. 1996; Liu et al. 1996) and on general deformable objects (Baraff and Witkin 1992; Metaxas and Terzopoulos 1992; Güdükbay et al. 1993).
- Particle models (Reeves 1983; Peachey 1986). These models have been intensively used to treat certain natural phenomena such as water (Fournier and Reeves 1986; Xu et al. 1997; van Wijk 1993; Goss 1990; Kass and Miller 1990), smoke (Chiba et al. 1994) and fire (Loke et al. 1992; Stam and Fiume 1995).

Likewise, there are groups that have developed their own symbolic languages for manipulating the equations that appear in this kind of systems (for example, Liu and Cohen 1995a).

In the simulation process, objects become masses with forces and torques acting on them. Motion arises applying the laws of classical mechanics. Therefore, the term simulation is frequently used instead of animation.

There are two approaches for using dynamics:

- *Direct dynamics*, where forces and torques are known and motion is obtained with minimal control over the system. Once initial conditions are given, the system evolves “alone”. It fits passive systems, without internal forces and torques
- *Inverse dynamics*, where motion is known and forces and torques are unknown and computed. It is best suited to motor systems that convert internal energies to time-dependent forces that produce their own motion. Here, the problem is the lack of realism: arbitrary motion without physical foundation can be generated

Various formulations have been adopted to obtain motion equations (Badler 1991):

- Newton-Euler (Gascuel and Gascuel 1994)
- Lagrange (Bruderlin and Calvert 1989; Vasilonikolidakis and Clapworthy 1991)
- Gibbs-Appell (Wilhelms 1987)
- D’Alembert (Isaacs and Cohen 1988)

Compared to purely kinematic ones, dynamic simulations have the advantage of giving great realism and reacting automatically to the environment (collisions, inertia, etc.). However, they pose other problems such as the control of the animation and the high computational cost of generating them. There have been two strong tendencies among the groups devoted to the application of dynamics to computer animation:

- Focus on developing new control schemes
- Attempts to reduce computational complexity.

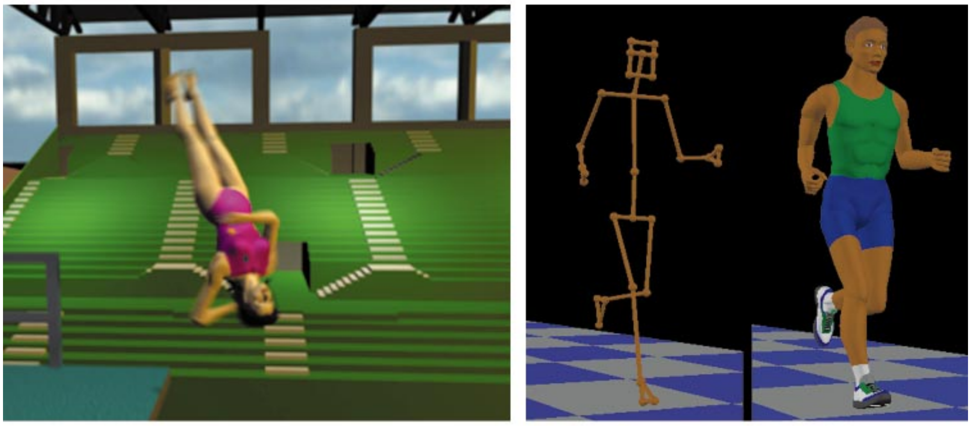
Even nowadays, the challenge is to obtain user-controlled realistic motion with small computation times.

3.2.2 Control in dynamic systems

The *control problem* (Wilhelms 1991) was initially tackled by translating the control necessities into a problem of applying forces: this is the so-called

6

Fig. 6. Synthetic “dressed” Marilyn by Miralab, University of Geneva



7

Fig. 7. Dynamic simulation examples of athletic behaviors by Hodgins et al. at the Graphics, Visualisation and Usability Center University of Georgia

preprocessing control. In most cases, it takes the form of springs and dampers. The method is capable of implementing objectives when the spring rest position coincides with the objective position. It can also consider joint limits and collisions if the simulated spring is made to compress when the limit is surpassed or when a collision occurs. The user can simply specify the objective or the path in a kinematic way. Systems of this kind are those of Barzel and Barr (1988), Wilhelms (1987), Wilhelms and Barsky (1985), Raibert and Hodgins (1991) and Liu et al. (1994). In general, these systems generate motion using different modules:

- Motion description modules where high-level control parameters such as step parameters, velocity and paths are considered
- Motor or control modules, which generate the guiding forces
- Dynamic modules, which establish motion equations and which usually have a numerical resolution submodule.

Bruderlin and Calvert (1989) present a hybrid kinematic/dynamic system to simulate human walking based on a deep kinematic knowledge with these three levels of control. Another example of simulation to generate human motion (running, cycling, jumping or swimming) by means of servo proportional/derivative forces is that of Hodgins et al. (1995), Wooten and Hodgins (1996) and Brogan et al. (1998) (Fig. 7). Lamouret et Gacuel (1996) use direct dynamics to simulate a trajectory constrained to a user-defined one.

Another kind of control is *matrix control*, where constraints are expressed in the form of equations. Isaacs and Cohen (1987, 1988) create a system to simulate articulated figures where the user is allowed to control the motion by means of kinematic constraints and behaviour functions that specify accelerations or forces as functions of time – a mixture of direct and inverse dynamics.

In all the systems we have mentioned, the resolution scheme is of the integral type. Once the motion equations have been established, they are solved with a numerical resolution system that integrates the equations in each time interval, moving forward in time. A more general vision of the control problem, at scene level, is that of Witkin and Kass (1988), who introduce the *space time* concept to refer to all the forces and to the values of all the degrees of freedom from the beginning to the end of the scene. The system automatically generates paths that have to fulfill objectives, mechanical laws (which are considered another type of constraint that ties forces and displacements) and minimize some functions (such as energy, softness, efficiency, etc.). This method is computationally expensive, and nonlinear optimization techniques have to be used (such as the conjugated gradient method). Systems of this type are those of Cohen (1992) and Cohen and Liu (1995b) who use a hybrid method between keyframe animation and optimization, and Liu et al. (1994) who propose the use of a hierarchical representation using wavelets for the functions that represent the temporal evolution of the generalised degrees of freedom.

3.2.3 Computational reduction techniques in dynamic systems

The search for real-time response has motivated the study of various simplifications in the dynamic models. Efficiency is the goal, to the detriment of precision. There are various approaches:

- *Recursive algorithms* suppose a relationship between forces and accelerations of consecutive joints, so that they do not have to simultaneously solve all the joints, but will do it in a recursive manner. This increases resolution time per joint and makes equations less intuitive. Their great advantage is that they succeed in passing from a computational complexity of $O(n^3)$ to $O(n)$, so that they are well suited to systems with many degrees of freedom. There are a number of recursive algorithms, such as those of Armstrong (1979), Featherstone (1987), Lathrop (1986), Hollerbach (1980) and Balafoutis and Patel (1991). Some implementation examples of this type of algorithm in computer animation systems are those of Armstrong and Green (1985) for the first one, McKenna and Zeltzer (1990) for the second, Vasilonikolidakis and Clapworthy (1991) for the fourth, and Rose et al. (1996) for the fifth algorithm.
- For their *sparse dynamics model*, Dworkin and Zeltzer (1993), propose discreet event simulation. Instead of integrating equations step by step, in regular time intervals, they propose a resolution in two steps. They make a first fast computation to estimate when the next collision will take place (first stage); then, the object is moved from one interaction to another, focusing computation in those events (second stage).
- Other simplifying ideas are those of van Overveld (1994), who suggests migrating from the rigid body dynamics to point mechanics to profit from current numerical techniques or considering some degrees of freedom kinematically driven through paths and leaving the others to the dynamic simulation (van Overveld and Ko 1994).
- Preston (1995) developes a distributed algorithm that tries to increase the motion generation speed by means of the use of *parallelization techniques*.

3.3 Task-oriented models: motor control

In the next generation of systems, namely the *task-oriented systems*, objects have environment perception, decision, action and communication abilities. Within these systems, a distinction should be made between:

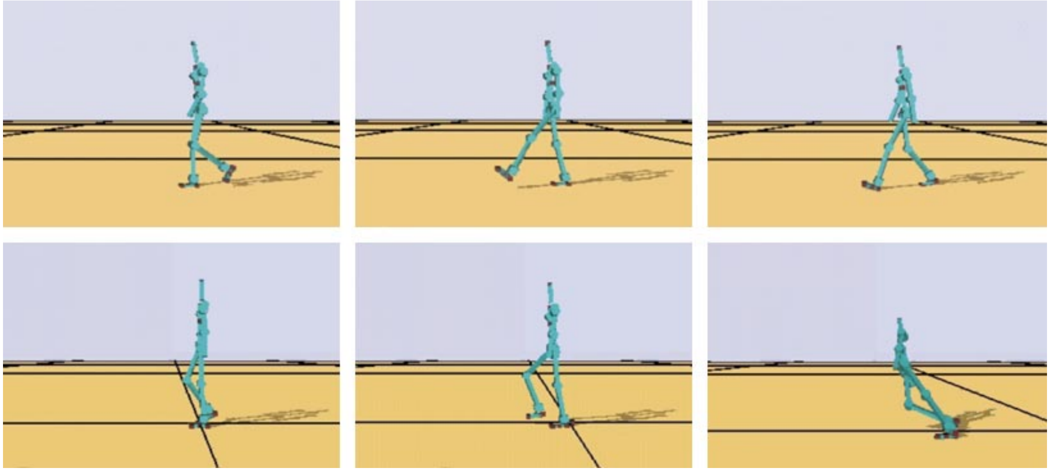


Fig. 8. Walking simulations using finite state machines. The first row corresponds to an open-loop unstable cycle. The second one shows the cycle with closed-loop feedback added (Laszlo et al. 1996)

- Motor or low-level control that provides basic actor movements and that can be physically based or simply heuristically based
- Motor planning or high-level control that connects perception and action.

In these systems, no paths are calculated for later refinement. Instead, the goal is to produce controllers that make decisions according to the information that they receive.

In this section, we focus on motor control. Control can be kinematic (Jung et al. 1994; Noser et al. 1995) or dynamic, as in the systems we have already commented on, but we must also consider other techniques.

3.3.1 Genetic programming techniques

The results of these techniques (Goldberg 1989; Davidor 1991) are control programs (Koza 1992). They need some starting variables and functions, a fitness function to measure individual fitting, a termination criterion, the number of individuals per generation and the maximum number of generations. It is a kind of artificial evolution in which only those individuals who fulfill the requirements survive (Gritz and Hahn 1995). Systems of this kind are those of Ngo and Marks (1993), Auslander et al. (1995) and Sims (1994) who uses these techniques to obtain not only motion, but also morphology of creatures.

3.3.2 Sensor-actuator networks (SAN)

Braitenberg (1984) plans some “thought experiments” using a balanced network of various kinds of nodes in order to demonstrate the possibility of obtaining intelligent behaviour. These ideas have been used by:

- Wilhelms and Skinner (1990), who work with solids capable of moving through 3D space. Combining sensors, effectors, arcs and nodes with adequate connections, they generate quite interesting movements with very little input information from the animator, using a network paradigm.
- Van de Panne and Fiume (1993) automatically synthesize the network. The user introduces the mechanical configuration of the system, augmented with some simple sensors and actuators. The system computes various forms of motion for each configuration. The searching process is stochastic. In order to reduce the searching space and to obtain balanced motion, van de Panne and Lamouret (1995) apply an external *momentum* to ensure a straight position during human locomotion (later, this momentum is eliminated).

3.3.3 Finite state machines

Tmovic and McGhee (1966) suggest the application of the automata theory of the finite state to the analysis and synthesis of engineering systems. This theory has also been applied to animation systems:

- Zeltzer (1982) and Zeltzer and Johnson (1993) model each joint controller as a finite-state machine. Over these machines, there are others (for each limb, for example) whose states are combinations of the former ones. The motor program (for walking, for example) is also a finite state machine whose states are made from local motor programs (LMPs) that have to be concurrently executed. The local motor program accedes to the joints by changing parameter values.
- Laszlo et al. (1996) use finite state machines with proportional-derivative (PD) controllers to generate an open-loop basic movement that is later disturbed to obtain stable motion (Fig. 8).

4 Behaviour-based modelling on animation systems

The last row of Table 2 shows behaviour modelling techniques that have been used, others which are nowadays being used and those that are starting to be used. In most cases, the final target is to obtain free synthetic actors in unpredictable virtual environments. Section 4.1 presents the evolution of these techniques and Sect. 4.2 shows the current trends.

4.1 Introduction and evolution of behaviour modelling

The various behaviour modelling techniques have evolved for some years now. The main changes can be summarised as follows:

- Techniques are moving from animations based on scripts (such as those in traditional animation) to those that do not need scripts.
- Current systems are normally modular and distributed, i.e. suitable for implementation in parallel architectures.
- Instead of using artificial intelligence (AI) tools, artificial life (Alife) elements are being introduced in behaviour modelling (Magnenat-Thalmann and Thalmann 1994).
- A structure has been created where low levels of animation (motor skills), high levels of animation (behaviour skills) and even other intermediate levels can be distinguished.
- Actors tend to be *autonomous*, *adaptive* and to have *learning skills*.
- A complex behaviour emerges from a combination of simple behaviours.

The most relevant works that show the stages of computer animation and behaviour are now described.

The work of Reynolds (1982, 1987) is the first important step in incorporating *behaviour* and *autonomy* concepts to classical computer animation systems. He proposes a "bottom-up" approach and designes a system where a global and complex behaviour emerges from a combination of several simple individual behaviours. Reynolds creates synthetic flocks of birds that avoid crashing into one another, maintain a constant velocity and remain within the flock. The animation he produces clearly shows these characteristics.

Another piece of pioneer work in creating synthetic actors, which is done by Magnenat-Thalmann and Thalmann (1987) is the "Human Factory" animation system, designed to reproduce synthetic actors who play the roles of famous stars already deceased, such as Marilyn Monroe or Humphrey Bogart. Their work provides solutions for modelling the skeleton and the body of the actors, and they focus on simple behaviours like grasping or expressing emotions facially. They conclude that the key features to obtain and use synthetic actors would be:

- To provide the actors with some knowledge about their environment
- To control their behaviour with an adequate level of abstraction
- To improve the quality of rendering for the animations
- To implement quick and reliable systems to design the actors physically.

The work done by Haumann and Parent (1988) is also very interesting. They present a computer animation system in which a complex movement is a result of the simulation of simple behaviour rules between locally related actors. Their object-oriented system is based on a message-passing mechanism. They are working to provide the actors not only with physical behaviour, but also with other behaviours that may express social or personal aspects.

After these first papers on behaviour and animation, an important need to define the current state of computer animation systems arises. They can no longer be considered systems that merely allow us to automate some parts of traditional animation and to accelerate parts of their production process. They are proper animation systems with several differentiated levels of work. Concepts like synthetic or autonomous actors and behaviour modelling are beginning to crystallize (as shown in column 3, Table 1). Among the works that define this point of inflexion, the following could be quoted: Zeltzer (1982), Zeltzer and Johnson (1993), Hegron et al. (1989), Renault et al. (1990) and Thalmann and Magnenat-Thalmann (1991).

From that moment on (columns 3 and 4, Table 1) a clear difference between physically based modelling and behaviour-based modelling is made, although sometimes both techniques are complementary. Physically based modelling, as stated in the previous section, emphasizes realistic aspects like elasticity, deformation or collision, as can be sampled in hair or clothes modelling. Behaviour-based modelling covers those internal aspects, some not yet well developed, like personality, social differences, perception or reaction. At this stage, the first papers devoted to behaviour modelling appear. The most relevant points are:

- Lethbridge and Ware (1989) present an animation system based on stimulus responses, i.e. perception-action. They state that, when an animated sequence is done, the actors are forced to behave as organisms responding to stimuli in their own local environment. The essence of the method is to “show” each actor how to behave within the environment and with each other. Maiocchi and Pernici (1990) present their computer animation system Pinocchio, which incorporates the ability to create an animation sequence with very little or no direction from the animator. As in movie making, where the director gives general instructions to the actors who express the details of how to act in the scene by themselves, the system allows the animator to specify the global scene, the restrictions and the various objects or actors in the scene (in terms of scripts). Coordination issues are solved autonomously by the actors through message passing.
- Wilhelms and Skinner (1990) state that behavioural animation is a means of automatic control motion where the objects or actors to be animated are or should be able to feel or perceive the environment and to decide their own movement following some rules. They describe a system in which a network with sensors as inputs and effectors as outputs connected by arcs and other nodes is responsible for the motion of objects (see sensor-actuator networks in Sect. 3.2.2). They conclude that future computer animation systems will have to combine low-level techniques (like locomotion) and high-level techniques (like interpretation or behaviour) in an intelligent fashion.
- McKenna et al. (1990) present an application with a *perception-action* approach. They describe an animation system composed of two levels that cooperate. A *sensor-motor level*, the low level, is responsible for the locomotion of an actor (a synthetic insect). Mathematical oscillators (rhythm), reflexes (feedback from spatial restrictions) and kinematic laws (already seen in the previous section) are involved in this process. The reflexes behave like sensors, detecting holes or obstacles, and reacting to them. The oscillators define coordinated gaits that control the locomotion of the six legs of the virtual insect, offering stability in motion according to the situation. A *reactive level*, the high level, interprets the environment in order to decide the actions that the low level should carry out, like setting speed or choosing the direction to move in.
- Beer (1990, 1995) and Beer et al. (1991, 1992) define the concept of *computational neuroethology* on the basis of the fact that animals with simple nervous systems give rise to complex behaviour when they try to adapt themselves to a changing environment. They apply this principle to an artificial hexapod insect, showing a 2D graphical simulation of its behaviour. They also apply it to robotics (Beer et al. 1997). For the simulation, they use a set of simulated sensors that allow it to know the state of the body (leg position) and the environment (food, other objects and energy). The possible behaviours are locomotion, turning, wall following and eating. The whole framework is based on a neural network where the nodes are neurons and the connections are activatory or inhibitory, and

the resulting behaviour is the imitation of an insect with a simple nervous system.

- Mah et al. (1994) develop a computer animation system to model behaviour, using the latest advances in expert systems and knowledge engineering. Their approach uses a control structure of several agents that interact through a reasoning process that uses an inference engine. Beardon and Ye (1995) propose a similar schema, using a production rule system to incorporate behaviour to computer animation. Cremer et al. (1996) define a methodology, split in several steps, to create a virtual scenario or environment and to create graphical simulations within it. When specifying the control of the actors, they propose a control schema that includes coordination of various agents, planning and high-level behaviours.

4.2 New trends in behaviour modelling

We now outline some of the more relevant and innovative works on behaviour modelling for computer animation. More details can be found in Badler et al. (1991), Trappl and Petta (1997) and Earnshaw et al. (1998).

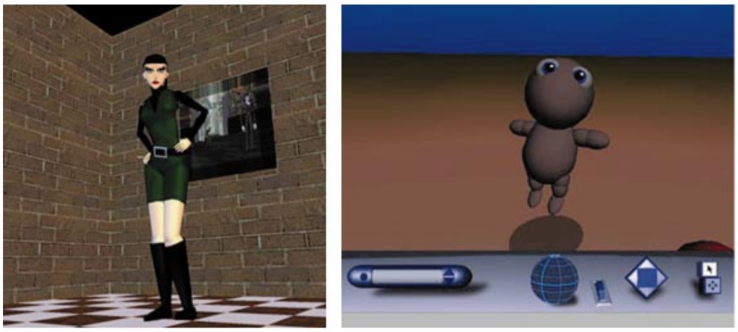
Maes (1990, 1994, 1995) presents a general schema for model behaviour, which can be applied to several fields, including robotics and computer animation (Brooks 1989, 1991). Their work has been a constant reference and a source of inspiration for other behaviour modelling systems in computer animation. The approach uses an perception-action schema as opposed to other classical approaches, which are based mostly on planning. They propose an action selection algorithm, based on the current situation and current goals. It evaluates the environment and acts during execution (as opposed to other existing algorithms, which need to be compiled and are therefore much less versatile and do not prove to be dynamic in a changing environment during execution). The resulting system is characterised by a behaviour network in which each node represents a concrete behaviour. Inside the network a constant energy is created, and based on current goals, situations and relations among behaviours, continuous competition between these behaviour nodes is maintained, ensuring correct action selection decisions each time.

The work presented by Tu and Terzopoulos (1994) (Fig. 9) shows a bottom-up approach, and uses a physical model to design actors (fish) and an environment for them. The synthetic animal has a locomotion model, a sensory system and a behaviour model. At the lowest level of abstraction, the fish is physically based on graphical models. The sensory system is responsible for perceiving a dynamic environment. At the highest level of abstraction, the behaviour system decides the most adequate behaviour or action to undertake at each iteration, and acts as a mediator between the perception system and a motor system. The latter is responsible for the locomotion of the artificial fish.

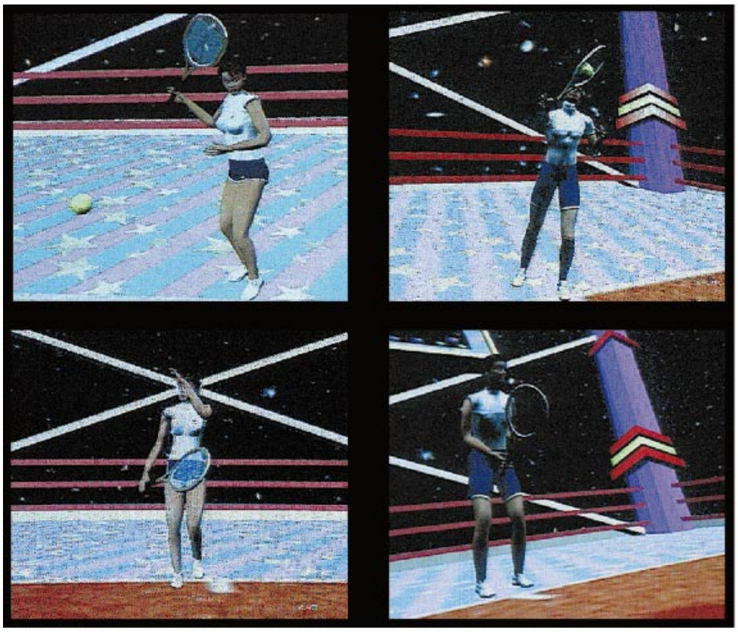
The work of Blumberg and Galyean (1995) and Blumberg (1997) focuses on the design of artificial creatures and virtual environments. Although the final goal of these papers is to provide an external direction of actors in real-time computer animation systems, they present a general behaviour model, based on perception and action selection for autonomous animated characters. The schema they propose defines three levels: perception, behaviour and motor. The main idea is (1) to satisfy a set of goals in a complex and dynamic environment, solving competition between goals, some of them concurrent, (2) to deal with errors or incompleteness during the perception stage and (3) to avoid dithering between behaviours.

Thalmann and colleagues have been working on complex systems that mix virtual reality, computer graphics and animation, with the special aims of obtaining great realism and of designing autonomous synthetic actors that imitate human behaviour (Boulic and Thalmann 1992; Carignan et al. 1992; Boulic et al. 1994; Thalmann and Magnenat-Thalmann 1991; Thalmann 1996; Kalra et al. 1998). Thalmann et al. (1997) present a system, currently under development, that simulates the artificial life of synthetic actors. On the basis of a perception-action schema, they implement virtual sensations (vision, tact and audition) and simulate actions in response to this perception such as locomotion (leg motion), grasping (with the hand) or “ball-in-air following” using synthetic vision. The system they present is an interesting alternative in behaviour modelling. It may be a solution to providing an actor with the necessary information to find a path, avoid obstacles, play or interact with other actors, construct an internal representation of the environ-

9



10



11

Fig. 9. “Jaws- shark stalking prey” (**a**) and “Two fish engage in mating ritual” (**b**) by Xiaoyuan Tu and colleagues (courtesy of Xiayuan Tu and DGP, Toronto)

Fig. 10a, b. “Sid” (**a**) and “Wendy” (**b**) created with the Java version of Improv running within a VRML browser by K. Perlin and A. Goldberg (IMPROV Project, Media Research Laboratory, New York University)

Fig. 11. “Anyone for Tennis?” directed by N. Magnenat-Thalmann and D. Thalmann (courtesy of Miralab, University of Geneva)

ment and learn or forget the kind of things humans usually do.

The work of Badler et al. (1993, 1997), the creators of the Jack copyright, is another example of a computer animation system that includes a behaviour model. They propose an agent-based architecture that presents two work levels. The lower level is called the sensor-control-action (SCA) while the higher level presents a well-defined, but still general, schema: parallel transition networks (PaT-Nets). The SCA produces local and adaptive movements. It consist of sensors, control nodes and actuators. Its basic behaviours are of the "to walk", "to go to", "to look at" type. The PaT-Nets allow the expression of more elaborate behaviour patterns than the previous basic ones. They are in fact automata, able to run in parallel and based on the current state of the environment, the current goal(s) and/or state of the system. The combination of SCA and PaT-Nets produces an adequate schema to define complex behaviour for synthetic actors.

We want to mention another computer animation system based on scripts (Goldberg 1997) that enables the creation of autonomous synthetic actors. The IMPROV project is being developed at New York University's Research Media Lab (Fig. 10) and it uses classical scripts for a given actor. These scripts are divided into groups. Each group contains all the scripts (or behaviours, we might say) that are mutually exclusive, in the sense that they are incompatible. However, in some cases, scripts belonging to different groups may be executed at the same time (meaning that a particular behaviour may have some scripts working in parallel, even if they are contradictory). Furthermore, autonomy is added to the system by allowing the scripts to have several different conditional executions, each of which has its own probability of success. The actors can also have some special personal characteristics and preferences, establishing their own personality. These attributes allow actors to be provided with criteria to decide the adequate alternative when dealing with such conditional scripts. The overall result is that the system uses conditional scripts in which the solution chosen is based on the criteria and personalities of the actors and on the given probabilities. By defining scripts, groups of scripts, personalities, criteria and probabilities, it is possible to obtain actors possessing personality and autonomy. The system is capable of producing different computer animations depending on these parameters and the interaction between scripts, i.e. actors.

Another research area on synthetic actors and virtual scenarios, is the study of theatre and the acting (art) of real actors or characters to apply these principles to the creation of believable synthetic actors. This is the case for Hayes-Roth (1995) and Hayes-Roth et al. (1997), who study the meaning of personality in a context where artificial agents behave like actors. From this point of view, they define concepts like *role*, *behaviour* and *improvisation* for synthetic actors.

Another concern in the design of synthetic actors is the fact that they should be believable, that their behaviour should seem natural and that they have to express emotions and use a natural language to communicate (Bates 1994; Loyall 1997).

The research work by Pina and Serón (Pina 1998a, b) proposes using classical artificial intelligence tools together with other techniques coming from the artificial life field. The goal is to design, first of all, a complex behaviour for synthetic actors, and second, changing scenarios for these actors. In order to make this possible, the actor (principal actor) is provided with decision, learning and adaptation capabilities. These characteristics are implemented with a fuzzy logic-based expert system, a neural network and a selection-action algorithm. The dynamic scenario is designed by using different objects: static and dynamic objects and extra actors. The latter can be any objects, living or not, and can have associated genetic algorithms that control their evolution in the scenario.

A new and interesting classification of actors in computer animation is proposed by Thalmann (1996) and Cavazza et al. (1998), where various classes of synthetic actors are distinguished:

- Virtual actors or participants represent the virtual animation of a person in real time, through virtual reality devices.
- Guided actors are actors directly manipulated by the animator or final user. Their motion does not have to be the same as that of the animator or user.
- Autonomous actors are able to generate their own behaviour. This implies that they have to identify the scenario in order to generate appropriate actions required by their behaviour. Among this type of actors, they distinguish

those that are passive and evolve in the environment without "touching anything" and those who interact actively with the environment and even communicate with other actors (interactive-perceptive actors).

This classification shows that autonomous actors need virtual sensors (Noser et al. 1995, Thalmann et al. 1997) to perceive the environment correctly, and behaviour modules to decide the actions to take on. Noser et al. (1996) and Molet et al. (1999) give an example of this approach. It is about an interactive game of tennis (Fig. 11) involving a guided actor and two interactive autonomous actors, Marilyn and a Referee, both provided with artificial visual capabilities. The simulation of the ball motion, gravity, and its interaction with the net is achieved by physically based techniques.

5 Reflections

Man possesses a remarkable ability to perceive the most subtle details of motion. For computer-generated motion to be convincing and give a lifelike impression, animated objects must behave in a natural manner.

Nowadays there are numerous applications of motion synthesis: cinematographic animation, virtual environments, videogames, ergonomics, rehabilitation, etc. Although the course embarked upon during the 1970s to get a computer to specify motion has been an extremely difficult one, the results attained are magnificent, and the future is looking bright, as this survey has tried to reflect. For example, at present, animation of certain characters' movements attempts to capture the subtleties of motion to express *personality* and *state of mind*, sketching out ever more credible models as a greater number of details are incorporated to the related algorithms. The number of work teams involved in the issues at hand is large and diversified.

Perhaps the matter still pending, in comparison with the advances made in motion and behaviour modelling, remains the design of user-friendly control systems. The difficulty resides in the design of control systems from the animator's viewpoint that do not require in-depth knowledge of behaviour algorithms and motion equations. In addition, problems increase as we go further, since synthetic motion becomes increasingly true to life when we take into account secondary elements that interact in response to the actions of the main elements.

The solid scientific and technical foundations developed during the last 30 years have rendered computer animation potentially adequate for a great number of entertainment, scientific and technological applications. Nonetheless, we still face challenges that will demand generating motion and behaviour with a *variety*, *expressiveness*, *interactivity* and *realism* that, as of yet, we can only find in reality.

6 Appendix

The following is a list of research groups, who are actually working on innovative areas of computer animation.

MIRALab, a research laboratory in virtual reality, computer animation and telepresence, was created in 1989 at the University of Geneva, and is directed by Professor Nadia Magnenat-Thalmann. This laboratory is specialised in modelling and animation of virtual human beings (e.g. Rendez-vous à Montréal). See:

http://miralabwww.unige.ch/.

The Center for Human Modelling and Simulation of the University of Pennsylvania is directed by N. Badler. The center works on human motion modelling and animation (Jack). The overall goal of the centre is the modelling and animation of Human Movement. This central topic guides a number of related research interests covering a broad scope from image synthesis to natural language interfaces. See:

http://www.cis.upenn.edu/∼hms/home.html.

The IMPROV Project, at New York University's Research Media Lab is building technologies to produce distributed responsive virtual environments in which human-directed avatars and computer-controlled agents interact in real time through a combination of procedural animation and behavioural scripting techniques. See:

http://www.mrl.nyu.edu/improv.

The Computer Graphics Lab (LIG) at the Swiss Federal Institute of Technology (EPFL) in Lausanne was founded in July 1988 by Professor Daniel Thalmann (its director). The laboratory is mainly involved in computer animation and virtual reality. Together with MIRALab (University of Geneva), LIG is especially well known for the creation and animation of virtual actors like synthetic Marilyn Monroe. Research at the Computer Graphics Laboratory (LIG) is oriented towards the virtual worlds, particularly the simulation of real-time virtual humans. See:

http://ligwww.epfl.ch/.

The group managed by P.D. Stroud in the Los Alamos National Laboratory, Technology and Safety Assessment Division, works on the design of intelligent actors for synthetic environments. See:

http://sgt-york.lanl.gov/homepages/stroud/icdoc/ic1.html

http://sgt-york.lanl.gov/homepages/stroud/icdoc/ic2.html

http://sgt-york.lanl.gov/homepages/stroud/icdoc/ic3.html

The *Virtual Theatre Project* aims to provide a multimedia environment in which users can interact with intelligent, automated actors, either in well-defined stories or in improvisational environments. Users themselves become actors by exercising high-level control over their own intelligent agents. These agents improvise to meet the user's goals on the basis of their knowledge, personalities and moods. See:

http://ksl-web.stanford.edu/projects/cait/

The Oz Project at the University of Carnegie-Mellon is developing technology and art to help artists to create high-quality interactive drama, based in part on AI technologies. In particular, this means building believable agents in dramatically interesting microworlds. See:

http://www-cgi.cs.cmu.edu/afs/cs.cmu.edu/project/oz/web/.

The Software Agents Group, at MIT Media Laboratory, works on developing agents, i.e. computational systems, that perceive, react and behave in dynamic and complex environments. One of their projects tries to create animated actors that live in 3D virtual worlds. See:

http://lcs.www.media.mit.edu/groups/agents/.

At the University of Georgia the Graphics, Visualisation and Usability Center (GVU), managed by J. Rossignac, works on several research projects involving fields such as visualisation, computer animation and virtual reality, collaborative design, usability, multimedia, cognition, digital culture, internet tools, education and future computing environments. See:

http://www.cc.gatech.edu/gvu.

The Graphics and Multimedia Research Lab of Simon Fraser University, Canada has developed the software package LifeForms, an innovative tool for 3D human figure animation, dance choreography, movement planning, game development, multimedia content creation and education. See:

http://www.cs.sfu.ca/research/groups/GMRL/projects/lifeforms.html.

IMAGIS (models, algorithms, geometry for graphics and image synthesis) is a team of the GRAVIR/IMAG Research Lab. This laboratory works mainly in modelling the physical behaviour of deformable objects: construction, simulation of movement, interactive manipulation, collision detection and response, simulation and control of articulated structures and particles systems. See:

http://www-imagis.imag.fr/.

The University of Toronto's Dynamic Graphics Project (DGP) is an interdisciplinary research laboratory within the Computer Science Department and the Computer Systems Research Institute. Research areas at DGP span a wide range of interests, including modelling (implicit models and interval methods, muscle models, representing levels of detail) and animation (physically based simulation, control and learning, hybrid kinematic/dynamic techniques). See:

http://www.dgp.utoronto.ca/DGP/.

The Human Figure Animation Project of Microsoft is working to make better and more realistic animation of humans for computer graphics. The work involves motion capture analysis and reuse, torque-minimal transitioning of motion capture, and most recently, deriving controllable animation through interpolation of motion-captured or hand-animated source environments. See:

http://www.research.microsoft.com/research/graphics/hfap/.

The Manchester Visualization Centre (formerly the Computer Graphics Unit), directed by W. T. Hewitt, has been working, among other things, on applying parallel computing to motion synthesis and cloth animation. Nowadays, work focuses on integrating interactive computer animation into multimedia presentations and developing new interactive motion synthesis tools that help an animator describe the potential for motion. See:

http://www.man.ac.uk.MVC/r.

The Perceptual Science Laboratory at the University of California at Santa Cruz is engaged in a variety of experimental and theoretical inquiries in perception and cognition. A major research area concerns speech perception by ear and eye, and facial animation. See:

http://mambo.ucsc.edu/.

References

Armstrong WW (1979) Recursive solution to the equations of motion of an *n*-link manipulator. Proceedings of the 5th World Congress on the Theory of Machines and Mechanism, pp 1343–1346

Armstrong WW, Green MW (1985) The dynamics of articulated rigid bodies for purposes of animation. Proceedings of Graphics Interface'85, Montreal, Canadian Information Processing Society, Toronto, Ontario, Canada, pp 407–415

Auslander J, Fukunaga A, Partovi H, Christensen J, Hsu L, Reiss P, Shuman A, Marks J, Ngo JT (1995) Further experience with controller-based automatic motion synthesis for articulated figures. ACM Trans Graph 14:311–336

Badler NI (1982) Human body models and animation IEEE Comput Graph Appl 2(9):6–7

Badler et al. (eds) (1991) Making them move: Mechanics, Control and Animation of Articulated Figures. Morgan Kauffmann, San Mateo, Calif.

Badler NI, Philipps CB, Webber BL (1993) Simulating humans. Oxford University Press, New York

Badler NI, Reich BD, Webber BL (1997) Towards personalities for animated agents with reactive and planning behaviours. In: Trappl R, Petta P (eds) Creating personalities for synthetic actors, Springer, Berlin Heidelberg New York, pp 43–57

Baecker RM (1969) Picture-driven animation. Proc Spring Joint Computer Conference, AFIPS Press, 34:273–288

Balafoutis CA, Patel RV (1991) Dynamic analysis of robot manipulators: a cartesian tensor approach. Kluwer Academic, Boston, USA

Baraff D, Witkin A (1992) Dynamic simulation of non-penetrating flexible bodies. SIGGRAPH'92, Comput Graph 26:303–308

Barr A (1984) Global and local deformations of solid primitives. SIGGRAPH'84, Comput Graph 18:21–30

Barr A (1987) Chair, "Topics in Physically-Based Modeling" Course Notes, Vol 16, ACM SIGGRAPH

Barr A, Currin B, Gabriel S, Hughes JF (1992) Smooth interpolation of orientations with angular velocity constraints using quaternions. Comput Graph 26:313–320

Bartels RH, Beatty JC, Barsky BA (1987) An introduction to splines for use in computer graphics and geometric modelling. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, Brian BA, series editor, Morgan Kaufmann, Los Altos, CA

Barzel R (1992) Physically based modelling for computer graphics. Academic Press, Inc., San Diego, Calif

Barzel R, Barr AH (1988) A modelling system based on dynamic constraints. SIGGRAPH'88, Comput Graph 22: 179–188

Bates J (1994) The role of emotion in believable agents. Commun ACM 37:122–125

Beardon C, Ye V (1995) Using behavioural rules in animation. In: R.A. Earnshaw and J.A. Vince (eds) Computer graphics: developments in virtual enviroments, Academic Press, London, pp 217–234

Beer RD (1990) Intelligence as adaptative behaviour. Academic Press

Beer RD (1995) A dynamical systems perspective on agent-environment interaction. Artif Intell 72:173–215

Beer RD, Chiel HJ, Sterling LS (1991) An artificial insect. Am Sci 79:444–452

Beer RD, Quinn RD, Chiel HJ, Larsson P (1992) A distributed neural network architecture for hexapod robot locomotion. Neural Comput 4:356–365

Beer RD, Quinn RD, Chiel HJ, Ritzmann RE (1997) Biologically inspired approaches to robotics. Commun ACM 40:31–38

Blumberg B (1997) Multi-level control for animated autonomous agents: do the right thing ... Oh, not that. In: Trappl R, Petta P (eds) Creating personalities for synthetic actors. Springer, Berlin Heidelberg New York, pp 74–82

Blumberg BM, Galyean TA (1995) Multilevel direction of autonomous creatures for real-time virtual environments. SIGGRAPH'95, Comput Graph Proc, Annual Conferences Series 1995, ACM 1993 SIGGRAPH Proc. pp 47–54

Bodenheimer B, Rose C, Rosenthal S, Pella J (1998) The process of motion capture: dealing with the data. Computer Animation and Simulation'97, Budapest, Hungary, Proceedings of the Eurographics Workshop. Thalman D and von de Panne M (eds)

Boulic R, Thalmann D (1992) Combined direct and inverse kinematic control for articulated figure motion editing. Comput Graph Forum 11:189–202

Boulic R, Magnenat-Thalmann N, Thalmann D (1990) A global human walking system with real-time kinematic personification. Visual Comput 6:344–358

Boulic R, Huang Z, Magnenat-Thalmann N, Thalmann D (1994) Goal-oriented design and correction of articulated rigure motion with the track system. Comput & Graph 18:443–452

Boulic R, Capin T, Huang Z (1995) The HUMANOID environment for interactive animation of multiple deformable human characters. EUROGRAPHICS'95, Comput Graph Forum 14:337–348

Boulic R, Mas R, Thalmann D (1995) Inverse kinetics for center of mass position control and posture optimization. In: Parker Y, Wilburg S (eds) Wokshop in Computing Series. Springer, Berlin Heidelberg New York, pp 234–249

Braitenberg V (1984) Vehicles: experiments in synthetic psychology. MIT Press, Cambridge, Mass.

Brogan DC, Metoyer RA, Hodgins JK (1998) Dynamically simulated characters in virtual environments. IEEE Comput Graph Appl 18:58–69

Brooks RA (1989) A robot that walks: emergent behaviours from a carefully evolved network. Neural Comput 1:253–262

Brooks RA (1991) New approaches to robotics. Science 253:1227–1232

Bruderlin A, Calvert TW (1989) Goal-directed, dynamic animation of human walking. Comput Graph 23:233–242

Bruderlin A, Williams L (1995) Motion signal processing. SIGGRAPH'95, Comput Graph Proc, Annual Conference Series 1995, ACM SIGGRAPH, pp 97–104

Bruderlin A, Teo CG, Calvert T (1994) Procedural movement for articulated figure animation. Comput & Graph 18:453–461

Burtnyk N, Wein M (1971) Computer-generated key-frame animation. J Soc Motion Picture and Television Enginees 80:149–153

Calvert TW, Chapman J, Patla A (1982) The simulation of human movement. Proceedings Graphics Interface '82:227–234

Calvert T, Bruderlin A, Dill J, Schiphorst T, Welman C (1993) Desktop animation of multiple human figures. IEEE Comput Graph Appl 13:18–25

Carignan M, Yang Y, Magnenat-Thalmann N, Thalmann D (1992) Dressing animated synthetic actors with complex deformable clothes. Comput Graph 26:99–104

Cavazza M, Earnshaw R, Magnenat-Thalmann N, Thalmann D (1998) Motion control of virtual humans. IEEE Comput Graph Appl 18:24–31

Chiba N, Muraoka K, Takahashi, H, Miura, M (1994) Two-dimensional visual simulation of flames, smoke and spread of fire. J Visualisation Comput Animat 5:37–53

Cohen M (1992) Interactive spacetime control for animation. SIGGRAPH'92, Comput Graph 26:293–302

Coquillart S (1990) Extended free-form deformation: a sculpturing tool for 3D geometric modelling. Comput Graph 24: 187–196

Coquillard S, Jancène P (1991) Animated free-form deformation: an interactive animation technique. SIGGRAPH'91, Comput Graph 25:23–26

Cremer J, Kearney J, Ko H (1996) Simulation and scenario support for virtual environments. Comput & Graph 20: 199–206

Davidor Y (1991) Genetic algorithms and robotics: a heuristic strategy for optimisation. World Scientific Series in Robotics and Intelligent Systems Harris CJ (ed)

Deussen O, Kobbelt L, Tücke P (1995) Using simulated annealing to obtain good nodal approximation of deformable bodies. In: Terzopoulos D, Thalmann D (eds) Computer Animation and Simulation'95 (Computer science series) Springer, Berlin Heidelberg New York, pp 30–43

Dworkin P, Zeltzer D (1993) A new model for efficient dynamic Simulation: Proceedings of the 4th Eurographics Animation and Simulation Workshop, pp 135–147

Earnshaw R, Magnenat-Thalmann N, Terzopoulos D, Thalmann D (1998) Computer animation for virtual humans. IEEE Comput Graph Appl 18:20–23

Eisert P, Girod B (1998) Analyzing facial expressions for virtual conferencing. IEEE Comput Graph Appl 18:70–77

Emering L, Boulic R, Balcisoy S, Thalmann D (1997) Real-time interactions with virtual agents driven by human action identification. First ACM Conference on Autonomous Agents'97, Marina del Rey, pp 476–477

Featherstone R (1987) Robot dynamics algorithms. Kluwer Academic, Boston, Mass.

Fournier A, Reeves WT (1986) A simple model of ocean waves. SIGGRAPH'86, Comput Graph 20:75–84

Gascuel JD, Gascuel MP (1994) Displacement constraints for interactive modelling and animation of articulated structures. Visual Comput 10:191–204

Geib C, Levison L, Moore MB, Sodajack: an architecture for agents that search for and manipulate objects. Technical Report MS-CIS-94-16 University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, PA, 1994

Girard M, Maciejewski AA (1985) Computational modelling for the computer animation of legged figures. SIGGRAPH'85, Comput Graph 19:263–270

Goldberg A (1997) IMPROV: a system for real-time animation of behaviour-based interactive synthetic actors. In: Trappl R, Petta P (eds) Creating personalities for synthetic actors. Springer, Berlin Heidelberg New York, pp 58–73

Goldberg DE (1989) Genetic algorithms in search, optimisation, and machine learning. Addison Wesley, Reading, Mass.

Goss ME (1990) A real-time particle system for display of ship wakes. IEEE Comput Graph Appl 10:30–35

Gourret JP, Magnenat-Thalmann N, Thalmann D (1989) Simulation of object and human skin deformations in a grasping task. SIGGRAPH'89, Comput Graph 23:21–30

Gritz L, Hahn K (1995) Genetic programming for articulated figure motion. J Visualisation Comput Animat 6:129–142

Güdükbay U, Özgüç B, Tokad Y (1993) An animation system for rigid and deformable objects. Comput & Graph 17:71–77

Haumann DR, Parent RE (1988) The behavioral test-bed: obtaining complex behaviour from simple rules. Visual Comput 4:332–347

Hayes-Roth B (1995) An architecture for adaptive intelligent agents. Artif Intell 72:329–365

Hayes-Roth B, Van Gent R, Huber D (1997) Acting in character. In; Trappl R, Petta P (eds) Creating personalities for synthetic actors. Springer, Berlin Heidelberg New York, pp 92–112

Hegron G, Palamidese P, Thalmann D (1989) Motion control in animation, simulation and visualization. Comput Graph Forum 8:347–352

Hoch M, Litwinowicz PC (1996) A semi-automatic system for edge tracking with snakes. Visual Comput 12:75–83

Hodgins JK, Wooten WL, Brogan DC, O'Brien JF (1995) Animating human athletics. SIGGRAPH'95, Comput Graph Proc, Annual Conference Series 1995, ACM SIGGRAPH, pp 71–78

Hollerbach J (1980) A recursive Lagrangian formulation of manipulator dynamics for robot manipulators. IEEE Trans Syst Man Cyber SMC-10:730–736

Hsu WM, Hughes JF, Kaufman H (1992) Direct manipulation of free-form deformations. SIGGRAPH'92, Comput Graph 26: 177–184

Isaacs PM, Cohen MF (1987) Controlling dynamic simulation with kinematics constraints, behaviour functions and inverse dynamics. SIGGRAPH'87, Comput Graph 21:215–224

Isaacs PM, Cohen MF (1988) Mixed methods for complex kinematics constraints in dynamic figure animation. Visual Comput 4:296–305

Jung MR, Badler N, Noma T (1994) Animated human agents with motion planning capability for 3D-space postural goals. J Visualization Comput Animat 5:225–246

Jüttler B (1994) Visualization of moving objects using dual quaternion curves. Comput & Graph 18:315–326

Kalra P, Magnenat-Thalmann N, Moccozet L, Sannier G, Aubel A, Thalmann D (1998) Real-time animation of realistic virtual humans. IEEE Comput Graph Appl 18:42–56

Kass M, Miller G (1990) Rapid, stable fluid dynamics for computer graphics. Comput Graph 24:49–57

Knowlton KC (1964) A computer technique for producing animated movies. Proc SJCC AFIPS Conference 25:67–87

Knowlton KC (1970) EXPLOR-A generator of images Proc 9th UAIDE Annual Meeting: pp 543–583

Kochanek D, Bartels R (1984) Interpolationg splines with local tension, continuity and bias control. Comput Graph 18:33–41

Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge, Mass.

Kunii TL, Gotoda H (1990) Singularity theoretical modelling and animation of garment wrinkle formation process. Visual Comput 6:326–336

Lamouret A, Gascuel MP (1996) Scripting interactive physically based motions with relative paths and synchronization. Comput Graph Forum 15:25–34

Laszlo J, Panne M van de, Fiume E (1996) Limit cycle control and its application to the animation of balancing and walking. SIGGRAPH'96, Comput Graph Proc, Annual Conference Series 1996, ACM SIGGRAPH, pp 155–162

Lathrop RH (1986) Constraint (closed-loop) robot simulation by local constraint propagation. Proceedings of 1986 IEEE International Conference on Robotics and Automation, San Francisco, 2:689–694

Lethbridge TC, Ware C (1989) A simple heuristically based method for expressive stimulus-response animation. Comput & Graph 13:297–303

Ling L, Damodaran M, Gay RKL (1996) A model for animating the motion of cloth. Comput & Graph 20:137–156

Liu Z, Cohen MF (1995a) An efficient symbolic interface to constraint based animation systems. In: Terzopoulos D, Thalmann D (eds) Computer Animation and Simulation'95, Proceedings of the Eurographics Workshop in Maastricht, The Netherlands (1995) Springer Verlag/Wien, pp 210–219

Liu Z, Cohen MF (1995b) Keyframe motion optimization by relaxing speed and timing. In: Terzopoulos D, Thalmann D (eds) Computer Animation and Simulation'95, Proceedings of the Eurographics Workshop in Maastricht, The Netherlands (1995) Springer Verlag/Wien, pp 144–153

Liu Z, Gortler SJ, Cohen MF (1994) Hierachical spacetime control. SIGGRAPH'94, Comput Graph Proc, Annual Conference Series, 1994, ACM SIGGRAPH, pp 35–42

Liu W, Ko MT, Chang RC (1994) An interactive approach to planning snake motion. Comput & Graph 18:537–542

Liu JD, Ko MT, Chang RC (1996) Collision avoidance in cloth animation. Vis Comput 12:234–243

Loke TS, Tan D, Seah HS, Er MH (1992) Rendering fireworks displays. IEEE Comput Graph Appl 12:33–43

Loyall AB (1997) Some requirements and approaches for natural language in a believable agent. In: Trappl R, Petta P (eds) Creating personalities for synthetic actors. Springer, Berlin Heidelberg New York, pp 113–119

Louchet J, Provot X, Crochemore D (1995) Evolutionary identification of cloth animation models. In: Terzopoulos D, Thalmann D (eds) Computer Animation and Simulation'95. Proceedings of the Eurographics Workshop in Maastricht, The Netherlands (1995) Springer Verlag/Wien, pp 44–54

LuoY, Perales López FJ, Villanueva Pipaon JJ (1992) An automatic rotoscopy system for human motion based on a biomechanic graphical model. Comput & Graph 16:355–362

Maes P (1990) Situated agents can have goals. In: Maes P (ed) Designing autonomous agents: theory and practice from biology to engineering and back. MIT, Elsevier Science, North Holland, Amsterdam, Netherlands, pp 49–70

Maes P (1994) Modeling adaptive autonomous agents. Artif Intell 1:135–162

Maes P (1995) Artificial life meets entertainment: lifelike autonomous agents. Commun ACM 38:108–114

Mah S, Calvert TW, Havens W (1994) A constraint-based reasoning framework for behavioural animation. Comput Graph Forum 13:315–324

Magnenat-Thalmann N, Thalmann D (1985) Computer animation: theory and practice. Springer, Tokyo

Magnenat-Thalmann N, Thalmann D (1990) Keyframe and painting systems. In: Tosiyasu LKunii (ed) Computer animation, theory and practice. Springer, Berlin Heidelberg New York, pp 41–60

Magnenat-Thalmann N, Thalmann D (eds) (1994) Artificial life and virtual reality. John Wiley, Chichester

Magnenat-Thalmann N, Laperrière R, Thalmann D (1988) Joint dependent local deformations for hand animation and object grasping. Proc Graphics Interface'88, Edmonton

Maiocchi R, Pernici B (1990) Directing an animated scene with autonomous actors. Visual Comput 6:359–371

Mas-Sanso R, Thalmann D (1994) A hand control and automatic grasping system for synthetic actors. EUROGRAPHICS'94, Comput Graph Forum 13:167–177

Mas-Sansó R, Boulic R, Thalmann D (1996) Cinética inversa generalizada con restricciones múltiples para el diseño interactivo de posturas balanceadas. VI Congreso Español de Informatica Grafica (CEIG'96), Valencia, Spain. Actas del Congreso, pp 243–269

McKenna M, Zeltzer D (1990) SIGGRAPH'90, Dynamic simulation of autonomous legged locomotion. Comput Graph 24:29–38

McKenna M, Pieper S, Zeltzer D (1990) Control of a virtual actor: the roach. 1990 Symposium on Interactive 3D Graphics. Comput Graph 24:165–174

Metaxas D, Terzopoulos D (1992) Dynamic deformation of solid primitives with constraints. Comput Graph 26:309–312

Meyer K, Applewhite H, Biocca F (1992) A survey of position trackers. Presence: teleoper virtual environ 1, 2: 173–200

Molet T, Aubel A, Capin T, Carion S, Lee E, Magnenat-Thalmann N, Noser H, Pandzic J, Sannier G, Thalmann D (1999) Ayone for Tennis? Presence, vol 8, n°2 pp 140–156

Ng HN, Grimsdale RL, Allen WG (1995) A system for modelling and visualization of cloth material. Comput & Graph 19:423–430

Ngo JT, Marks J (1993) Spacetime constraint revisited. SIGGRAPH'93, Comput Graph Proc, Annual Conference Series, 1993, ACM SIGGRAPH pp 343–350

Noser H, Renault O, Thalmann D, Magnenat-Thalmann N (1995) Navigation for digital actors based on synthetic vision, memory and learning. Comput & Graph 19:7–19

Noser H, Pandzic IS, Capin TK, Magnenat Thalmann N, Thalmann D (1996) Playing games through the virtual life network. Proc Artificial Life V Nara, Japan, 114–121

Overveld CWAM van (1994) A simple approximation to rigid body dynamics for computer animation. J Visualization Comput Animat 5:17–36

Overveld CWAM van, Ko H (1994) Small steps for mankind: towards a kinematically driven dynamic simulation of curved path walking. J Visualization Comput Animat 5: 143–165

Panne M van de, Fiume E (1993) Sensor-actuator networks. SIGGRAPH'93, Comput Graph Proc, Annual Conference Series 1993, ACM SIGGRAPH, pp :335–342

Panne M van de, Lamouret A (1995) Guided optimization for balanced locomotion. In: Terzopoulos D, Thalmann D (eds) Computer Animation and Simulation'95. Proceedings of the Eurographics Workshop in Maastricht, The Netherlands (1995) Springer Verlag/Wien, pp 165–177

Parke FJ (1972) Animation of faces Proceedings ACM Annual Conference, vol 1

Paul RP (1981) Robot manipulators: mathematics, programming and control. MIT Press, Cambridge, Mass.

Peachey DR (1986) Modelling waves and surf. SIGGRAPH'86, Comput Graph 20:65–86

Pina A, Serón FJ (1998a) Modelling behaviour and motion control in computer animation with intelligent objects. Proceedings of the International Symposium on Engineering of Intelligent Systems/EIS'98, University of La Laguna, Tenerife, Spain. pp 177–184

Pina A, Serón FJ (1998b) Obtaining autonomous computer animated actors using fuzzy expert systems and neural networks. Proceedings of the 2nd International Conference on Computer Graphics and Artificial Intelligence, University of Limoges, Limoges, France, pp 115–125

Platt SM, Badler NI (1981) Animating facial expressions. Comput Graph 15:245–252

Preston M (1995) Parallel spacetime animation. In: Computer Animation and Simulation '95 Proceedings of the Eurographics Workshop in Maastricht, The Netherlands (1995) Terzopoulos D, Thalmann D (Eds) Springer-Verlag, Wien, pp 187–196

Preston M, Hewitt WT (1994) Animation using NURBS. Comput Graph Forum 13:229–241

Raibert MH, Hodgins JK (1991) Animation of dynamic legged locomotion. SIGGRAPH'91, Comput Graph 25:349–358

Reeves W (1983) Particle-systems – a technique for modelling a class of fuzzy objects. SIGGRAPH'83, Comput Graph 17: 359–376

Renault O, Magnenat-Thalmann N, Thalmann D (1990) A vision-based approach to behavioural animation. J Visualization Comput Animat 1:18–21

Reynolds, CW (1982) Computer animation with scripts and actors. SIGGRAPH'82, Comput Graph 16:289–296

Reynolds CW (1987) Flocks, herds, and schools: a distributed behavioural model. Comput Graph 21:25–34

Rose C, Guenter B, Bodenheimer B, Cohen MF (1996) Efficient generation of motion transitions using spacetime constraints. SIGGRAPH'96, Comput Graph Proc, Annual Conference Series 1996 ACM SIGGRAPH, pp 147–154

Rose C, Cohen MF, Bodenheimer B (1998) Verbs and adverbs: multidimensional motion interpolation. IEEE Comput Graph Appl 18:32–40

Sederberg TW, Parry SR (1986) Free-form deformations of solid geometric models. Comput Graph 20:151–160

Shoemake K (1985) Animating rotation with quaternion curves. Comput Graph 19:245–254

Sims K (1994) Evolving virtual creatures. SIGGRAPH'94, Comput Graph Proc, Annual Conference Series 1994 ACM SIGGRAPH, pp 15–22

Snibbe S (1995) A direct manipulation interface for 3D computer animation. EUROGRAPHICS'95, Comput Graph Forum 14:271–283

Spencer-Smith T, Wyvill G (1989) Four dimensional splines for motion control in computer animation. State-of-the-art in computer animation. In: Magnenat-Thalmann N, Thalmann D (eds) Proceedings of Computer Animation'89, Springer-Verlag, Tokyo 1989, pp 153–167

Stam J, Fiume E (1995) Depicting fire and other gaseous phenomena using diffusssion processes. SIGGRAPH'95, Comput Graph Proc, Annual Conference Series 1995, ACM SIGGRAPH, pp 129–136

Terzopoulos D, Platt J, Barr A, Fleischer K (1987) Elastically deformable models. SIGGRAPH'87, Comput Graph 21: 205–214

Thalmann D (1996) A new generation of synthetic actors: the real-time and interactive perceptive actors. Pacific Graphics'96 Taipeh, Taiwan, pp 200–219

Thalmann D, Magnenat Thalmann N (1987) The direction of synthetic actors in the film *Rendez-vous à Montréal*. IEEE Comput Graph Appl 7:9–19

Thalmann D, Magnenat Thalmann N (1991) Complex models for animating synthetic actors. IEEE Comput Graph Appl 11:32–44

Thalmann D, Noser H, Huang Z (1997) Autonomous virtual actors based on virtual sensors. In: Trappl R, Petta P (eds) Creating personalities for synthetic actors. Springer, Berlin Heidelberg New York, pp 25–42

Tmovic R, McGhee RB (1966) A finite state approach to the synthesis of bioengineering control system. IEEE Trans Hum Factors Electronics HFE-7:65–69

Trappl R, Petta P (eds) (1997) Creating personalities for synthetic actors. Springer, Berlin Heidelberg New York

Tu X, Terzopoulos D (1994) Artificial fishes: physics, locomotion, perception, behaviour. SIGGRAPH'94, Comput Graph Proc, Annual Conference Series 1994, ACM SIGGRAPH, pp 42–48

Unuma M, Anjyo K, Takeuchi R (1995) Fourier principles for emotion-based human figure animation. SIGGRAPH'95, Comput Graph Proc, Annual Conference Series 1995, ACM SIGGRAPH, pp 91–96

Vasilonikolidakis N, Clapworthy GJ (1991) Inverse lagrangian dynamics for animating articulated models. J Visualization Comput Animat 2:106–113

Volino P, Magnenat-Thalmann N (1995) Collision and self-collision detection: efficient and robust solutions for highly deformable surfaces. In: Terzopoulos D, Thalmann D (eds) Computer Animation and Simulation'95. Proceedings of the Eurographics Workshop in Maastricht, The Netherlands (1995), Springer Verlag/Wien, pp 55–65

Volino P, Magnenat-Thalmann N, Jianhua S, Thalmann D (1996) An evolving system for simulating clothes on virtual actors. IEEE Comput Graph Appl 16:42–51

Waters K (1988) The computer synthesis of expressive three-dimensional facial character animation. PhD Thesis, Middlesex University, Faculty of Art and Design, Cat Hill Barnet, Herts, UK

Webber B, Badler N, Di Eugenio B, Geib C, Levison L, Moore M. The Institut for Research in Cognitive Science Instructions, Intentions and Expectations IRCS Report 94–01, University of Pennsylvania, Philadelphia, PA 19104-6228, January 1994

Wijk JJ van (1993) Flow visualization with surface particles. IEEE Comput Graph Appl 13:18–24

Wilhelms J (1987) Using dynamic analysis for the animation of articulated bodies. IEEE Comput Graph Appl 7:11–22

Wilhelms J (1991) Dynamic experiences. In: Badler NI, Barsky BA, Zeltzer D (eds) Making them move: mechanics, control and animation of articulated figures. Morgan Kaufmann, San Mateo Calif., pp 265–279

Wilhelms J, Barsky BA (1985) Using dynamic analysis for the animation of articulated bodies such as humans and robots. Proceedings of Graphics Interface'85, Canadian Information Processing Society, Toronto, Ontario, Canada, pp 97–104

Wilhelms J, Skinner R (1990) A notion for interactive behavioural animation control. IEEE Comput Graph Appl 10: 14–22

Witkin A, Kass M (1988) Spacetime constraints. Comput Graph 22:159–168

Witkin A, Popovic Z (1995) Motion warping. SIGGRAPH'95, Comput Graph Proc, Annual Conference Series, 1995, ACM SIGGRAPH, pp 105–108

Wooten WL, Hodgins JK (1996) Animation of human diving. Comput Graph Forum 15:3–13

Xu Y, Su C, Qi D, Li H, Liu S (1997) Physically based simulation of water currents and waves. Comput & Graph 21:277–280

Zhao J, Badler N (1994) Inverse kinematics positioning using nonlinear programming for highly articulated figures. ACM Trans Graph 13:313–336

Zeltzer D (1982) Motor control techniques for figure animation. IEEE Comput Graph Appl 2:53–59

Zeltzer D, Johnson B (1993) Virtual actors and virtual environments: defining, modelling and reasoning about motor skills. In: MacDonald L, Vince J (eds) Interacting with virtual environments. John Wiley, Chichester, England

EVA CEREZO is a member of the Advanced Computer Graphics Group at the University of Zaragoza, Spain. She is currently a PhD candidate in Computer Science. Her research interests include physically based modelling and medical imaging. She obtained a BS in Physics in 1990 and a MsC in Nuclear Physics from the University of Zaragoza in 1992.

ALFREDO PINA is currently a PhD candidate in Computer Science at the Public University of Navarra, Pamplona, Spain. His research interests include artificial life, artificial intelligence and behavior modelling for computer animation. He obtained a BS in Computer Science from Bordeaux University, France, in 1986, and a MsC in KBS from Heriot-Watt University, Scotland, in 1989. He is a collaborator of the Advanced Computer Graphics Group (GIGA) at the University of Zaragoza, Spain.

FRANCISCO J. SERON is an Associate Professor of Computer Science at the Technical School of Engineering at the University of Zaragoza. He received a Physical Science Degree from the University of Zaragoza in 1977 and a PhD from the same university in 1984. At present, he is the Head of the Advanced Computer Graphics Group within the Computer Science Department. His research interests include simulation of natural phenomena, illumination engineering and virtual reality.