Physical, Behavioral, and Sensor-Based Animation

Daniel Thalmann
Computer Graphics Lab
Swiss Federal Institute of Technology (EPFL)
CH 1015 Lausanne, Switzerland
E-mail: thalmann@lig.di.epfl.ch
WWW: http://ligwww.epfl.ch

1. Introduction

An important part of the current animation consists in simulating the real world. To achieve a simulation, the animator has two principal techniques available. The first is to use a model that creates the desired effect. A good example is the growth of a green plant. The second is used when no model is available. In this case, the animator produces "by hand" the real world motion to be simulated. Until recently most computer-generated films have been produced using the second approach: traditional computer animation techniques like keyframe animation, spline interpolation, etc. Automatic motion control techniques have been proposed, but they are strongly related to mechanics-based animation and do not take into account the behavior of characters. However, high level animation involving human beings and animals may be produced using behavioral and perception models.

1.1 Classification of methods

Magnenat Thalmann and Thalmann [1] propose a classification of computer animation scenes involving synthetic actors both according to the method of controlling motion and according to the kinds of interactions the actors have. A *motion control method* specifies how an actor is animated and may be characterized according to the type of information to which it is privileged in animating the synthetic actor. For example, in a keyframe system for an articulated body, the privileged information to be manipulated is joint angles. In a forward dynamics-based system, the privileged information is a set of forces and torques. The nature of privileged information for the motion control of actors falls into three categories: geometric, physical and behavioral, giving rise to three corresponding categories of motion control method.

- The first approach corresponds to methods heavily relied upon by the animator: rotoscopy, shape transformation, parametric keyframe animation. *Synthetic actors are locally controlled.* Methods are normally driven by geometric data. Typically the animator provides a lot of geometric data corresponding to a local definition of the motion. Inverse kinematic methods may be also considered as being in this category.
- The second way guarantees a realistic motion by using kinematics and dynamics. The problem with this type of animation is controlling the motion produced by simulating the physical laws which govern motion in the real world. The animator should provide physical data corresponding to the complete definition of a motion. The physical laws involved are mainly those of mechanics. As trajectories and velocities are obtained by solving equations, we may consider *actor motions as globally controlled.* Functional methods based on biomechanics are also part of this class.
- The third type of animation is called behavioral animation and takes into account the relationship between each object and the other objects. Moreover the control of animation may be performed at a task- level, but we may also consider *the actor as an autonomous creature.* In fact, we will consider as a behavioral motion control method any method consisting in driving the behavior of this actor by providing high-level directives indicating a specific behavior without any other stimulus.

2. Motion Control Methods of Articulated Bodies

2.1 Skeleton definition

Most animated characters are structured as articulated bodies defined by a skeleton. When the animator specifies the animation sequence, he/she defines the motion using this skeleton. A **skeleton** [2] is a connected set of segments, corresponding to limbs, and joints. A **joint** is the intersection of two segments, which means it is a skeleton point where the limb which is linked to the point may move. The angle between the two segments is called the **joint angle**. A joint may have at most three kinds of position angles: flexing, pivot and twisting.

2.2 Kinematics methods for skeleton animation

In this group of methods, the privileged information is of a geometric or kinematics nature. Typically, motion is defined in terms of coordinates, angles and other shape characteristics or it may be specified using velocities and accelerations, but no force is involved. Among the techniques based on geometry and kinematics, we will discuss performance animation, keyframing, morphing, inverse kinematics and procedural animation. Although these methods have been mainly concerned with determining the displacement of objects, they may also be applied in calculating deformations of objects.

Skeleton animation consists of animating joint angles. Among the best-known methods in the category of geometric motion control methods for animating skeletons, we may consider **rotoscopy**, using sensors to provide coordinates of specific points of joint angles of a real human for each frame. **Keyframe systems** are typical of systems that manipulate angles; for example, to bend an arm, it is necessary to enter into the computer the elbow angle at different selected times. Then the software is able to find any angle at any time using for example interpolating splines.

Morphing is a technique which has attracted much attention recently because of its astonishing effects. It is derived from shape transformation and deals with the metamorphosis of an object into another object over time. While three-dimensional object modeling and deformation is a solution to the morphing problem, the complexity of objects often makes this approach impractical. The difficulty of the three-dimensional approach can be effectively avoided with a two-dimensional technique called image morphing. Image morphing manipulates two-dimensional images instead of three-dimensional objects and generates a sequence of inbetween images from two images. Image morphing techniques have been widely used for creating special effects in television commercials, music videos, and movies. The problem of image morphing is basically how an inbetween image is effectively generated from two given images. More detailed processes for obtaining an inbetween image are described by Wolberg [3]. Lee and Shin [4] has given a good survey of digital warping and morphing techniques.

2.2.1 Forward and inverse kinematics

The **forward kinematics** problem consists in finding the position of end point positions (e.g. hand, foot) with respect to a fixed-reference coordinate system as a function of time without regard to the forces or the moments that cause the motion. Efficient and numerically well-behaved methods exist for the transformation of position and velocity from joint-space (joint angles) to Cartesian coordinates (end of the limb). Parametric keyframe animation is a primitive application of forward kinematics.

The use of **inverse-kinematics** [5] permits direct specification of end point positions. Joint angles are automatically determined. This is the key problem, because independent variables in a synthetic actor are joint angles. Unfortunately, the transformation of position from Cartesian to joint coordinates generally does not have a closed-form solution. However, there are a number of special arrangements of the joint axes for which closed-form solutions have been suggested in the context of animation [6].

A higher level of specification of kinematics motion is based on the use of **constraints**. The animator impose a limb end to stay at a specified location or to follow a predefined trajectory. Badler et al. [7] have introduced an iterative algorithm for solving multiple constraints using inverse kinematics. In their system, the user has to specify also the precedence of each constraint in case they cannot all be simultaneously satisfied.

3. Dynamics

3.1 Dynamic Simulations

Kinematic-based systems are generally intuitive and lack dynamic integrity. The animation does not seem to respond to basic physical facts like gravity or inertia. Only modeling of objects that move under the influence of forces and torques can be realistic. Forces and torques cause linear and angular accelerations. The motion is obtained by the dynamic equations of motion. These equations are established using the forces, the torques, the constraints and the mass properties of objects. A typical example is the motion of an articulated figure which is governed by forces and torques applied to limbs.

Methods based on parameter adjustment are the most popular approach to dynamics-based animation and correspond to *non-constraint methods* . There is an alternative: the *constraint-based methods*: the animator states in terms of constraints the properties the model is supposed to have, without needing to adjust parameters to give it those properties.

3.2 Non-Constraint-Based Methods

Non-constraint methods have been mainly used for the animation of articulated figures. There are a number of equivalent formulations which use various motion equations:

- the Newton–Euler formulation
- the Lagrange formulation
- the Gibbs–Appell formulation
- the D'Alembert formulation

These formulations are popular in robotics and more details about the equations and their use in computer animation may be found in [8]. The *Newton-Euler formulation* [9] is based on the laws governing the dynamics of rigid bodies. The procedure in this formulation is to first write the equations which define the angular and linear velocities and accelerations of each link and then write the equations which relate the forces and torques exerted on successive links while under this motion. The equations of motion for robots can be derived through the application of the *Lagrange's equations* of motion for nonconservative systems. Based on this theory, Armstrong et al. [10] use a recursive method to design a near real-time dynamics algorithm and implement it in a prototype animation system. Wilhelms and Barsky [11] use the **Gibbs–Appel formulation** for their animation system Deva; however, the cost of solving for accelerations is prohibitively expensive (cost of $O(n^4)$). The **D'Alembert's principle of virtual work** states that if a system is in dynamic equilibrium and the bodies are allowed to move a small amount (virtual displacement) then the sum of the work of applied forces, the work of internal forces will be equal and opposite to the work of changes in momentum. Isaacs and Cohen [12] use the D'Alembert formulation in their DYNAMO system. Also, Arnaldi et al. [13] have produced a dynamics-based animation sequence consisting of an actress' arm, where the hand reaches a point from a rest position and then successively draws letters O and M from this point.

3.3 Constraint-based Methods

Isaacs and Cohen [14] discuss a method of constraint simulation based on a matrix formulation. Joints are configured as *kinematic constraints*, and either accelerations or forces can be specified for the links. Isaacs and Cohen also propose an integration of direct and inverse kinematics specifications within a mixed method of forward and inverse dynamics simulation. More generally, an approach to imposing and solving geometric constraints on parameterized models was introduced by Witkin et al. [15] using *energy constraints*. Using *dynamic constraints*, Barzel and Barr [16] build objects by specifying geometric constraints; the models assemble themselves as the elements move to satisfy the constraints. Once a model is built, it is held together by constraint forces. Platt and Barr [17] extend dynamic constraints to flexible models using reaction constraints and optimization constraints. Witkin and Kass [18] propose a new method, called *Spacetime Constraints*, for creating character animation. In this new approach, the character motion is created automatically by specifying *what* the character has to be, *how* the motion should be performed, what the character's *physical structure* is, what physical *resources* are available to the character to accomplish the motion. The problem to solve is a problem of constrained optimization.

Figure 1 shows an example of dynamics-based motion.

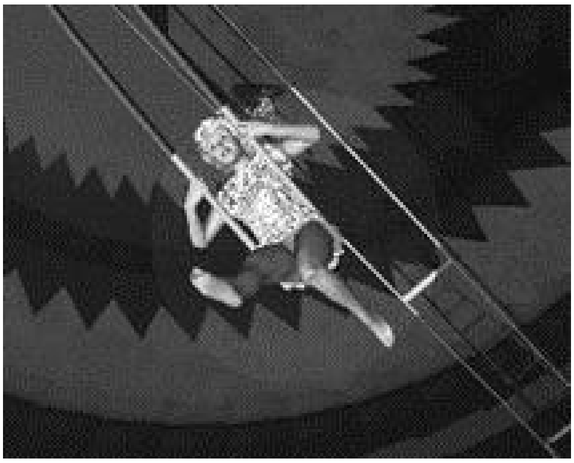

Figure 1. A motion calculated using dynamic simulation

3.4 Physics-based Deformations

Realistic simulation of deformations may be only performed using physics-based animation. The most well-known model is the Terzopoulos elastic model [19]. In this model, the fundamental equation of motion corresponds to an equilibrium between internal forces (inertia, resistance to stretching, dissipative force, resistance to bending) and external forces (e.g. collision forces, gravity, seaming and attaching forces, wind force). Gourret et al. [20] propose a finite element method to model the deformations of human flesh due to flexion of members and/or contact with objects. The method is able to deal with penetrating impacts and true contacts.

4. Task-level

4.1 Task-level Animation

According to Lozano-Perez's [21] description, task planning may be divided into three phases:

1) World modelling: it consists mainly of describing the geometry and the physical characteristics of the objects and the object.
2) Task specification: a task specification by a sequence of model states using a set of spatial

relationships [22] or a natural language interface is the most suitable and popular [23 24].

3) Code Generation: several kinds of output code are possible: series of frames ready to be recorded, value of parameters for certain keyframes, script in an animation language or a command-driven animation system.

In each case, the correspondence between the task specification and the motion to be generated is very complex. In the next sections, we consider two essential tasks for a synthetic actor: walking and grasping.

4.2 Walking

For many years there has been a great interest in natural gait simulation. According to Zeltzer [25], the gait cycle is usually divided into a stance phase, during which the foot is in contact with the ground, and a swing phase, where the leg is brought forward to begin the stance phase again. Each arm swings forward with the opposite leg and swings back while the opposite leg is in its stance phase. For implementing such a cycle walk, Zeltzer describes a walk controller invoking eight local motor programs (LMP): left swing, left stance, right swing, and right stance, which control the actions of the legs, hips, and pelvis; and four other LMPs that control the swinging of the arms. Girard and Maciejewski [26] use inverse kinematics to interactively define gaits for legged animals. Although Girard's model [27] also incorporates some dynamic elements for adding realism, it is not a truly dynamic approach. Also Bruderlin and Calvert [28] propose a hybrid approach to the human locomotion which combines goal-oriented and dynamic motion control. Knowledge about a locomotion cycle is incorporated into a hierarchical control process. McKenna and Zeltzer [29] describe an efficient forward dynamic simulation algorithm for articulated figures which has a computational complexity linear in the number of joints. Decomposition of the locomotion determines forces and torques that drive the dynamic model of the legs by numerical approximation techniques.To individualize human walking, Boulic et al. [30] propose a model built from experimental data based on a wide range of normalized velocities. The model is structured on two levels. At a first level, global spatial and temporal characteristics (normalized length and step duration) are generated. At the second level, a set of parameterized trajectories produce both the position of the body in space and the internal body configuration. The model is based on a simple kinematic approach designed to preserve the intrinsic dynamic characteristics of the experimental model. Figure 2 shows examples of walking sequences.

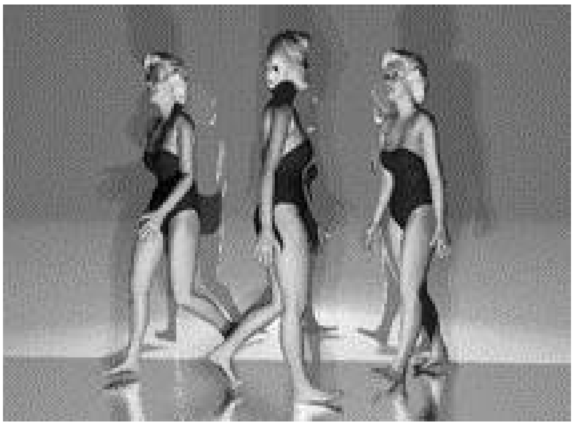

Figure 2. Walking sequence

4.3 Grasping

In the computer animation field, interest in human grasping appeared with the introduction of the synthetic actors. Magnenat Thalmann et al. [31] describe one of the first attempts to facilitate the task of animating actors' interaction with their environment. However, the animator has to position the hand and decide the contact points of the hand with the object. Figure 3 shows an example.

Figure 3. Grasping in the film "Rendez-vous à Montréal"

Rijpkema and Girard [32] presents a full description of a grasping system that allows both, an automatic or an animator chosen grasp. The main idea is to approximate the objects with simple primitives. The mechanisms to grasp the primitives are known in advance and constitute what they call the knowledge database. Recently, Mas and Thalmann [33] have presented a hand control and automatic grasping system using an inverse kinematics based method. In particular, their system can decide to use a pinch when the object is too small to be grasped by more than two fingers or to use a two-handed grasp when the object is too large (see [34] for more details).

Figure 4 shows an example of object grasping scene.

Figure 4. Object grasping scene

5. Behavioral animation

5.1 Introduction

Behavioral animation corresponds to modeling the behavior of characters, from path planning to complex emotional interactions between characters. In an ideal implementation of a behavioral animation, it is almost impossible to exactly play the same scene twice. For example, in the task of walking, everybody walks more or less the same way, following more or less the same laws. This is the "more or less" which will be difficult to model. And also a person does not walk always the same way everyday. If the person is tired, or happy, or just got some good news, the way of walking will appear slightly different. So in the future, another big challenge is open for the computer animation field: to model human behavior taking into account social differences and individualities.

Reynolds [35] studied in details the problem of group trajectories: bird flocks, herds of land animals and fish schools. This kind of animation using a traditional approach (keyframe or procedural laws) is almost impossible. In the Reynolds approach, each bird of the flock decide itself its trajectory without animator intervention. Reynolds introduces a distributed behavioural model to simulate flocks of birds, herds of land animals, and schools of fish. The simulated flock is an elaboration of a particle system with the simulated birds being the particles. A flock is assumed to be the result of the interaction between the behaviours of individual birds. Working independently, the birds try both to stick together and avoid collisions with one another and with other objects in their environment. The animator provides data about the leader trajectory and the behaviour of other birds relatively to the leader (e.g. minimum distance between actors). A computer-generated film has been produced by symbolic using this distributed behavioural model: *Breaking the ice.* Haumann and Parent [36] describe behavioural simulation as a means to obtain global motion by simulating simple rules of behaviour between locally related actors. Lethebridge and Ware [37] propose a simple heuristically-based method for expressive stimulus-response animation. They model stimulus-response relationships using "behaviour functions" which are created from simple mathematical primitives in a largely heuristic manner. Wilhelms [38] proposes a system based on a network of sensors and effectors. Ridsdale [39] proposes a method that guides lower-level motor skills from a connectionist model of skill memory, implemented as collections of trained neural networks. We should also mention the huge literature about autonomous agents which represents a background theory for behavioral animation. More recently, genetic algorithms were also proposed to automatically generate morphologies for artificial creatures and the neural systems for controlling their muscle forces. Another approach for behavioral animation is based on timed and parameterized **L-systems** [40] with conditional and pseudo stochastic productions. With this production-based approach a user may create any realistic or abstract shape, play with fascinating tree structures and generate any concept of growth and life development in the resulting animation.

5.2 Perception through Virtual Sensors

In a typical behavioral animation scene , the actor perceives the objects and the other actors in the environment, which provides information on their nature and position. This information is used by the behavioral model to decide the action to take, which results in a motion procedure. In order to implement perception, virtual humans should be equipped with visual, tactile and auditory sensors. These **virtual sensors** should be used as a basis for implementing everyday human behaviour such as visually directed locomotion, handling objects, and responding to sounds and utterances. For synthetic audition [41], one needs to model a sound environment where the synthetic actor can directly access positional and semantic sound source information of audible sound events. Simulating the haptic system corresponds roughly to a collision detection process. But, the most important perceptual subsystem is the vision system as described in the next section.

5.3 Virtual vision

The concept of synthetic vision was first introduced by Renault et al. [42] as a main information channel between the environment and the virtual actor. Reynolds [43] more recently described an evolved, vision-based behavioral model of coordinated group motion. Tu and Terzopoulos [44] also proposed artificial fishes with perception and vision. In the Renault method, each pixel of the vision input has the semantic information giving the object projected on this pixel, and numerical information giving the distance to this object. So, it is easy to know, for example, that there is a table just in front at 3 meters. The synthetic actor perceives his environment from a small window in which the environment is rendered from his point of view. As he can access z buffer values of the pixels, the color of the pixels and his own position he can locate visible objects in his 3D environment.

More recently, Noser et al. [45] proposed the use of an octree as the internal representation of the environment seen by an actor because it offers several interesting features. The octree has to represent the visual memory of an actor in a 3D environment with static and dynamic objects. Objects in this environment can grow, shrink, move or disappear. To illustrate the capabilities of the synthetic vision system, the authors have developed several examples: the actor going out of a maze, walking on sparse foot locations and playing tennis.

In the ALIVE system [46], the virtual world is inhabited by inanimate objects as well as agents. Agents are modeled as autonomous behaving entities which have their own sensors and goals and which can interpret the actions of the participant and react to them in "interactive-time". Agents in ALIVE use virtual sensors. The most commonly used virtual sensor in the ALIVE system works by shooting a number of rays out in a 2D plane across an arc of a specified angle. It records, for each ray, the closest point of intersection with another object (which may be an inanimate object, another agent or the user) as well as properties of that object. The ALIVE system creates a "special" 3D agent in the environment to represent the user. The position and state of that agent are based on the information computed by the vision system on the basis of the camera image of the user. Thus, the artificial agents can sense the user using the same virtual sensors that they use to detect objects and other agents. The special agent is not rendered in the final image, instead the live video image of the user is composited with the computer graphics.

We are currently investigating the mixing of autonomous actors with actors animated using sensors. As an example, we have produced a fighting between a real person represented by a synthetic character and an autonomous actor (see Figure 5). The motion of the real person is captured using a Flock of Birds. The gestures are recognized by the system and the information is transmitted to the virtual actor who is able to react to the gestures and decide which attitude to do.

Figure 5. Fighting between an autonomous actor and an actor controlled with a Flock of Birds

As shown in Figure 6, we also used a camera to introduce an autonomous real-time synthetic Marilyn inside our lab.

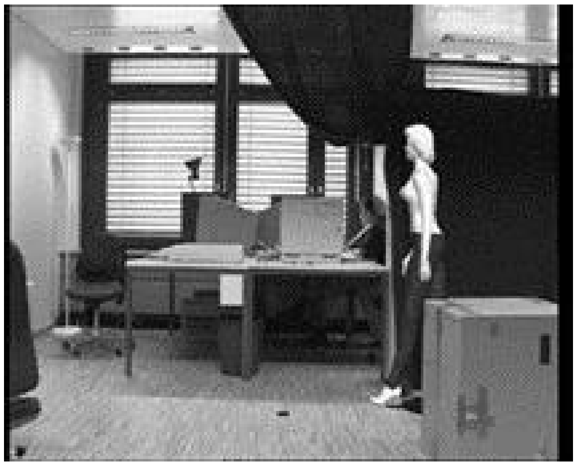

Figure 6. Marilyn visits our lab in real-time

6. References

[1] Magnenat Thalmann, N. and Thalmann, D. 1991. Complex Models for Animating Synthetic Actors, IEEE Computer Graphics and Applications, Vol.11, September.

[2] Badler NI and Smoliar SW (1979) Digital Representation of Human Movement, ACM Computing Surveys, March issue, pp.19-38.

[3] Wolberg G. 1990 "Digital Image Warping", IEEE Computer Society Press.

[4] Lee, S. and Shin S.Y. 1995. Warp Generation and Transition Control in Image Morphing in: Magnenat Thalmann, N. and Thalmann D. (eds) Interactive Computer Animation, Prentice Hall.

[5] Hollerbach JM, Sahar G (1983) Wrist-Partitioned, Inverse Kinematic Accelerations and Manipulator Dynamics, Intern. Journal

of Robotics Research, Vol.2, No4, pp.61-76.

[6] Badler NI, Korein JD, Korein JU, Radack GM and Brotman LS (1985) Positioning and Animating Figures in a Task-oriented Environment, The Visual Computer, Vol.1, No4, pp.212-220.

[7] Badler NI et al. (1986) Multi-Dimensional Input Techniques and Articulated Figure Positioning by Multiple Constraints, 1986 Workshop on Interactive 3D Graphics, Chapel Hill, North Carolina

[8] Thalmann D (1990) Robotics Methods for Task-level and Behavioral Animation, in: Thalmann D (ed) Scientific Visualization and Graphics Simulation, John Wiley, Chichester, UK, pp.129-147.

[9] Orin D, McGhee R, Vukobratovic M and Hartoch G (1979) Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler methods, Mathematical Biosciences, Vol.31, pp.107-130.

[10] Armstrong WW, Green M and Lake R (1987) Near real-time Control of Human Figure Models, IEEE Computer Graphics and Applications, Vol.7, No 6, pp.28-38

[11] Wilhelms J and Barsky B (1985) Using Dynamic Analysis to Animate Articulated Bodies such as Humans and Robots, in: N.Magnenat-Thalmann and D.Thalmann (Eds) Computer-generated Images, Springer, pp.209-229.

[12] Isaacs PM and Cohen MF (1987) Controlling Dynamic Simulation with Kinematic Constraints, Bahvior Functions and Inverse Dynamics, Proc. SIGGRAPH'87, Computer Graphics, Vol.21, No4, pp.215-224

[13] Arnaldi B, Dumont G, Hégron G, Magnenat-Thalmann N, Thalmann D (1989) Animation Control with Dynamics, in: Magnenat-Thalmann N, Thalmann D (eds) State-of-the-Art in Computer Animation, Springer, Tokyo, pp.113-124.

[14] Isaacs PM, Cohen MF (1988) Mixed Methods for Complex Kinematic Constraints in Dynamic Figure Animation, The Visual Computer, Vol. 4, No6, pp.296-305.

[15] Witkin A, Fleischer K, Barr A (1987) Energy Constraints on Parameterized Models, Proc. SIGGRAPH'87, Computer Graphics, Vol.21, No4, pp.225-232

[16] Barzel R, Barr AH (1988) A Modeling System Based on Dynamic Constraints, Proc. SIGGRAPH '88, Computer Graphics, Vol.22, No4, pp.179-188

[17] Platt JC, Barr AH (1988) Constraint Method for Flexible Models, Proc. SIGGRAPH '88, Computer Graphics , Vol. 22, No4 , pp.279-288.

[18] Witkin A, Kass M (1988) Spacetime Constraints, Proc. SIGGRAPH '88, Computer Graphics, Vol.22, No4 , pp.159-168.

[19] Terzopoulos, D. Platt, J.C. Barr, A.H., and Fleischer, K. 1987. Elastically Deformable Models, Proc.SIGGRAPH'87, Computer Graphics, Vol. 21 No 4, pp.205-214

[20] Gourret, J.P. Magnenat-Thalmann, N. and Thalmann, D. 1989. Simulation of Object and Human Skin Deformations in a Grasping Task, Proc. SIGGRAPH '89, Computer Graphics, Vol. 23, No 3, pp. 21-30

[21] Lozano-Perez (1982) Task Planning in: Brady M (Ed.) Robot Motion: Planning and Control, MIT Press, Cambridge, Mass.

[22] Popplestone RJ, Ambler AP and Bellos IM (1980) An Interpreter for a Language for Describing Assemblies, Artificial Intelligence, Vol.14, pp.79-107

[23] Drewery K, Tsotsos J (1986) Goal Directed Animation using English Motion Commands, Proc. Graphics Interface '86, pp.131-135

[24] Badler NI (1989) Artificial Intelligence, Natural Language, and Simulation for Human Animation, in: Magnenat-Thalmann N, Thalmann D (Eds) State-of.the-Art in Computer Animation, Springer, Tokyo, pp. 19-32

[25] Zeltzer (1982) Motor Control Techniques for Figure Animation, IEEE Computer Graphics and Applications , Vol. 2, No9 , pp.53-59.

[26] Girard M and Maciejewski AA (1985) Computational Modeling for Computer Generation of Legged Figures, Proc. SIGGRAPH '85, Computer Graphics, Vol. 19, No3, pp.263-270

[27] Girard M (1987) Interactive Design of 3D Computer-Animated Legged Animal Motion, IEEE Computer Graphics and Applications, Vol. 7, No 6, pp.39-51.

[28] Bruderlin A, Calvert TW (1989) Goal Directed, Dynamic Animation of Human Walking, Proc. SIGGRAPH '89, Computer Graphics, Vol. 23, No3, pp.233-242

[29] McKenna M, Zeltzer D (1990) Dynamic Simulation of Autonomous Legged Locomotion, Proc. SIGGRAPH '90, Computer Graphics, Vol. 24, No 4, pp.29-38.

[30] Boulic R, Magnenat-Thalmann N, Thalmann D (1990) A Global Human

Walking Model with real time Kinematic Personification, The Visual Computer, Vol.6, No6, pp.344-358.

[31] Magnenat-Thalmann, Laperrière, R., and Thalmann, D. 1988. Joint-Dependent Local Deformations for Hand Animation and Object Grasping, Proc. Graphics Interface'88, Edmonton, pp.26-33

[32] Rijpkema, H. and Girard, M. 1991. Computer Animation of Knowledge-based Grasping, Proc. SIGGRAPH '91, Computer Graphics, vol. 25, N. 4, pp:339-348.

[33] Mas, R. and Thalmann D. 1994. A Hand Control and Automatic Grasping System for Synthetic Actors, Proceedings of Eurographic'94, pp.167-178.

[34] Magnenat-Thalmann N, Thalmann D (1996) The Simulation of Virtual Humans (course), Proc. Graphicon 96, St Petersburg.

[35] C.Reynolds, Flocks, Herds, and Schools: A Distributed Behavioral Model, Proc.SIGGRAPH '87, Computer Graphics, Vol.21, No4, 1987, pp.25-34.

[36] Haumann DR, Parent RE (1988) The Behavioral Test-bed: Obtaining Complex Behavior from Simple Rules, The Visual Computer, Vol.4, No 6, pp.332-347

[37] Lethebridge TC and Ware C (1989) A Simple Heuristically-based Method for Expressive Stimulus-response Animation, Computers and Graphics, Vol.13, No3, pp.297-303

[38] Wilhelms J. 1990. A "Notion" for Interactive Behavioral Animation Control, IEEE Computer Graphics and Applications , Vol. 10, No 3 , pp.14-22

[39] Ridsdale G. 1990. Connectionist Modelling of Skill Dynamics, Journal of Visualization and Computer Animation, Vol.1, No2, pp.66-72.

[40] Noser, H. Turner, R., and Thalmann, D. 1992. Interaction between L-systems and Vector Force-Field, Proc. Computer Graphics International '92, Tokyo, pp.747-761.

[41] Noser, H. and Thalmann, D. 1995. Synthetic Vision and Audition for Digital Actors, Proc. Eurographics '95, Maastricht, August 1995, pp.325-336.

[42] Renault, O. Magnenat-Thalmann, N. and Thalmann, D. 1990. A Vision-Based Approach to Behavioural Animation, Visualization and Computer Animation Journal, John Wiley, Vol.1, No1, 1990, pp.18-21.

[43] Reynolds C.W. (1993) An Evolved, Vision-Based Behavioral Model of Coordinated Group Motion, in: Meyer J.A. et al. (eds) From Animals to Animats, Proc. 2nd International Conf. on Simulation of Adaptive Behavior, MIT Press, pp.384-392.

[44] Tu X., Terzopoulos D., Artificial Fishes: Physics, Locomotion, Perception, Behavior, Proc. SIGGRAPH '94, Computer Graphics, 1994, pp.42-48.

[45] Noser, H. Renault, O. Thalmann, D. and Magnenat Thalmann, N. 1995. "Navigation for Digital Actors based on Synthetic Vision, Memory and Learning", Computers and Graphics, Pergamon Press, Vol.19, No1, pp.7-19.

[46] Maes P, Darrell T, Blumberg B, Pentland A, The ALIVE system: Full-Body Interaction with Autonomous Agents, Proc. Computer Animation '95, IEEE Computer Society Press, 1995, pp.11-18.