MOTION CONTROL OF SYNTHETIC ACTORS: AN INTEGRATED VIEW OF HUMAN ANIMATION

Nadia Magnenat-Thalmann
Centre Universitaire d'Informatique
University of Geneva, Switzerland

Daniel Thalmann
Computer Graphics Lab
Swiss Federal Institute of Technology, Switzerland

Abstract

This paper explains the ideal concepts that must be part of a system for synthetic actor animation. After a brief introduction to the role of synthetic actors, five major steps to the motion control of these actors are discussed: positional constraints and inverse kinematics, dynamics, impact of the environment, task planning and behavioral animation.

1. The synthetic actor approach

Three-dimensional human modeling and animation have existed for more than 15 years [Badler,1982; Magnenat-Thalmann and Thalmann, 1985]. However emphasis has been placed on separate aspects of this animation. For example, a great deal of research has been done on the modeling and animation of the body [Badler and Smoliar, 1979; Kroyer, 1986; Forest et al. 1986], on facial animation [Parke, 1974; Platt and Badler, 1981; Pearce et al., 1986; Hill et al., 1988; Lewis and Parke, 1987; Waters, 1987; Magnenat-Thalmann et al., 1988b] and hand animation [Catmull, 1972; Badler and Morris, 1982; Magnenat-Thalmann et al. 1988]. More recently, researchers have incorporated mechanical aspects factors into the animation of articulated bodies [Armstrong and Green, 1985; Armstrong and Green, 1985b; Wilhelms and Barsky, 1985; Wilhelms, 1987; Girard, 1987; Isaacs and Cohen, 1987]. Also task level factors have been studied by several researchers [Zeltzer, 1982; Zeltzer, 1985; Badler et al., 1985; Badler et al. 1987]. Other authors have tried to animate more realistic characters, from an image synthesis point of view, but using primitive methods like rotoscopy or image-based keyframe animation. The synthetic actor approach corresponds to an integration of all methods, allowing the creation of 3D characters, with the appearance of real human characters.

Ideal synthetic actors should satisfy the following criteria:

- they should have the appearance of real persons
- their behavior should be similar to that of real persons
- they should have their own personality: two different actors should have different personalties, i.e. different reactions to the same situation

- they should be directed by task level commands
- they should be conscious of their environment
- they should at least be able to walk, speak, have emotions, and grasp objects
- their body and face should be naturally deformed during motions

Existing people, dead or alive can be recreated as synthetic actors or synthetic actresses. But fictitious people may also be created in this way.

Research in this area implies the development of techniques:

- for improving the physical aspects of the actors: shapes, colors, textures, reflectances
- for improving the motion of limbs and their deformation during motion
- for improving facial expressions and their animation
- for specifying the tasks to be performed
- for implementing tools for automatic motion control

Fig.1 shows complete diagram of the components of the Human Factory system presented together with the relations between these components.

This paper emphasizes the motion control of synthetic actors.

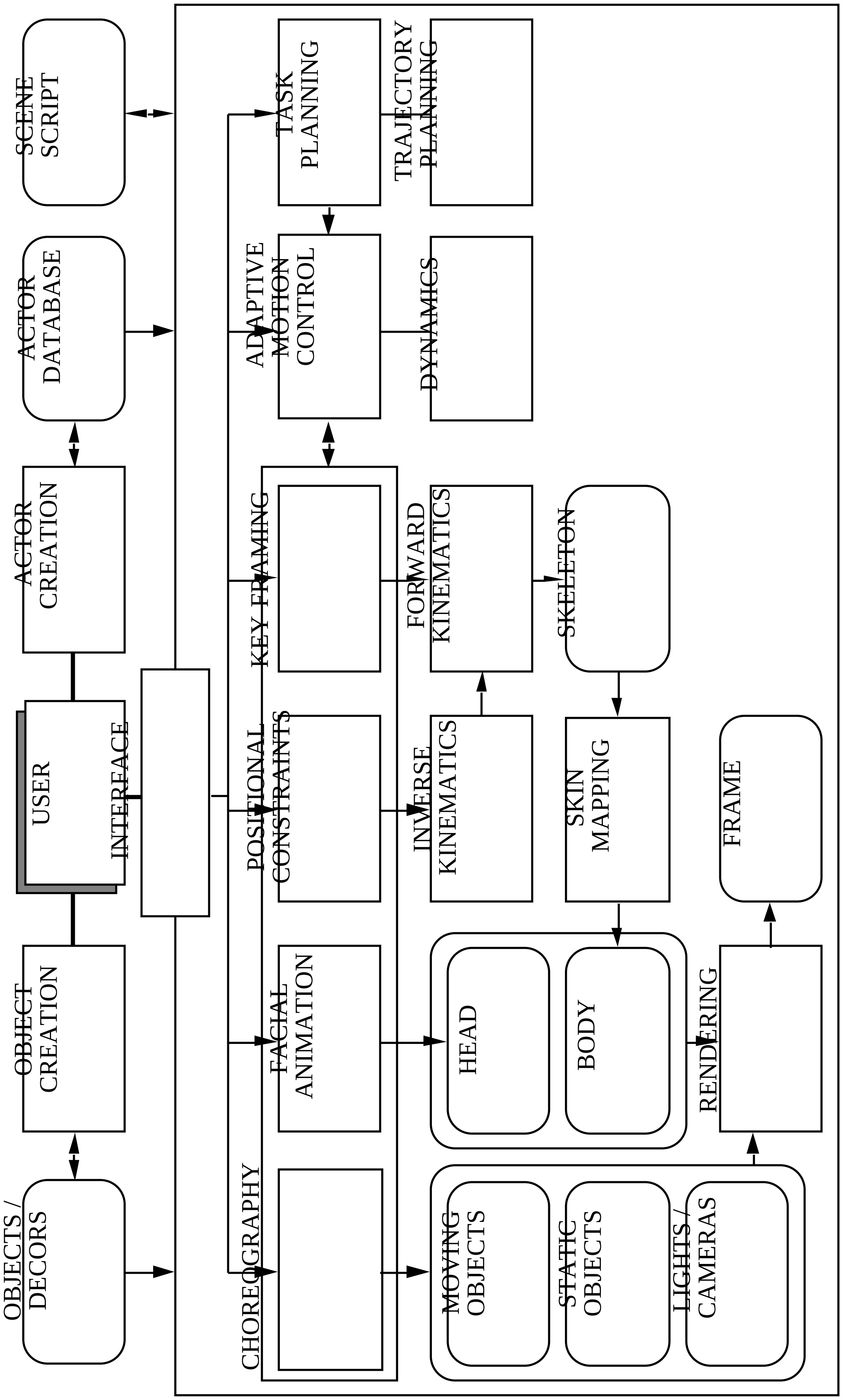

Fig.1 An integrated system

2. Motion control: from inbetweening to automatic control

The most common method in computer-assisted animation, called keyframe animation [Burtnyk and Wein, 1971], consists of the automatic generation of intermediate frames, called inbetweens, obtained by interpolating a set of key-frames supplied by the animator. A way of producing better images is to interpolate parameters of the model instead of the object itself. This is a popular technique called parametric keyframe animation [Parke, 1974]. In a parameter model, the animator creates keyframes by specifying the appropriate set of parameter values, parameters are then interpolated and images are finally individually constructed from the interpolated parameters. For example, to bend an arm of a synthetic actor, 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. Inbetween values are generally calculated using cubic splines.

In future animation systems, based on synthetic actors, motion control will be automatically performed using A.I. and robotics techniques. In particular, motion will be planned at a task level and computed using physical laws. In the next sections, five steps to automatic motion control are described.

4. First step: positional constraints and inverse kinematics

Consider the important problem of limb positioning, e.g.: what are the angle values for the shoulder, elbow and wrist if the hand has to reach a certain position and orientation in space? The problem involves the determination of the joint variables given the position and the orientation of the end of the hand with respect to the reference coordinate system. This is the key problem, because independent variables in a human being are joint variables; this problem is well-known in robotics and is called the inverse-kinematics problem. In a typical animation system based on inverse kinematics, the animator specifies discrete positions and motions for end parts; then the system computes the necessary joint angles and orientations for other parts of the body to put the specified parts in the desired positions and through the desired motions. Such an approach works well for simple linkages. However, the inverse kinematic solutions to a particular position become numerous and complicated, when the number of linkages increases. Let us have an example, it is not difficult to determine how much to bend an elbow and a wrist to reach an object with the hand. It is much more difficult if we bring into play the rotation of the shoulder and the flexion of fingers. The transformation problem from Cartesian coordinates has no closed-form solution in general. 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 [Badler et al., 1985; Girard and Maciejewski, 1985; Girard, 1987; Korein and Badler, 1982; Forsey and Wilhelms, 1988].

In order to make a synthetic actor sit down on a chair (see Fig.2), for example, it is necessary to specify the relevant constraints on the feet, on the pelvis and on the hands. A system which allows to specify only one constraint at a time is not a very efficient way to solve this problem. Badler et al. [1987] 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 the event case they cannot all be simultaneously satisfied.

Fig.2 The synthetic actress Eglantine

A simple algorithm solving the positional constraint problem has been implemented in our system [Boisvert et al., 1989]. The animator may impose constraints at the hands, the feet and the pelvis levels. The position and orientation of the hand or the feet may be specified in the local coordinate system attached to the limb (arm or leg), or in the actor system or the world system. A constraint may be a fixed position/orientation or a 6D trajectory. Tools are available for constructing constraints as functions of the actor environment and his envelop (e.g. contact of the foot and the floor). In order to solve the constraints, the system makes use of the position and orientation of the pelvis and the trunk angles (vertebrae and clavicles) for finding the origin of the hips and the shoulders. It then calculates the limb angles required to reach the intended position. In the case where no solution exists, the intended position is projected on the volume of moving of the arm (leg). The skeleton has seven degrees of freedom at the arm (leg) level and the constraint has six degrees (position/orientation). Since the model is redundant from a kinematics point-of-view, this implies the existence of an infinity of solutions to reach the intended position. One solution consists of minimizing the angle variation of the angle between the leg (arm) and the foot (hand). It is also possible to have the user select the solution by giving an opening parameter. The position/orientation/opening constraint allows to select a unique solution from the arm's (or the leg's) seven degrees of freedom. Other criteria such as the collision of the limb with an object may play a role in the selection of the solution.

The key framing technique and the positional constraints may be considered as the low level commands of an animation system. The higher level commands may produce keyframes and joint constraints. The animator must have access to the various levels of the hierarchy in order to be able to do the fine-tuning of the actor motion.

5. Second step: motion control using dynamics

A more complex, but more realistic approach is based on dynamics. The motion of a synthetic actor is governed by forces and torques applied to limbs. Two problems may be considered: the direct-dynamics problem and the inverse-dynamics problem. The direct-dynamics problem consists of finding the trajectories of some point as the end effector with regard to the forces or the torques that cause the motion. The inverse-dynamics problem is much more useful and may be stated as follows: given a trajectory as well as the forces to be exerted at the manipulator tip, find the torques to be exerted at the joints so as to move it in the desired manner. For a synthetic actor, it is possible to compute the time sequence of joint torques required to achieve the desired time sequence of positions, velocities and accelerations using various methods.

As discussed by Arnaldi et al. [1989], three main factors lead to introduce dynamics in animation control :

. dynamics frees the animator from the description of the motion due to the physical properties of the solid objects.
. reality of natural phenomena is better rendered.
. bodies can react automatically to internal and external environmental constraints: fields, collisions, forces and torques.

Techniques based on dynamics have already been used in computer animation [Armstrong and Green, 1985; Armstrong and Green, 1985b; Wilhelms and Barsky, 1985; Wilhelms, 1987; Isaacs and Cohen, 1987], but only for simplified and rigid articulated bodies with few joints, geometrical bodies (cylinders) and without any deformation. The use of the dynamics in an animation system of articulated bodies like the human body, provide several important disadvantages:

First, the animator does not think in terms of forces or torques to apply to a limb or the body in order to perform a motion. The design of a specific user interface is essential.

Another problem of the dynamics is the amount of CPU time required to solve the motion equations of a complex articulated body using numerical methods. It considerably reduces the possibility of interaction of the system with the user. Only very short sequences have been produced, because of the lack of complete specification for complex motions and because of the CPU time required for certain methods.

Moreover, although dynamics-based motions are more realistic, they are too regular, because they do not take into account the personality of the characters. It is unrealistic to think that only the physical characteristics of two people carrying out doing the same actions make these characters different for any observer. Behavior and personality of the human beings are also an essential cause of the observable differences.

6. Third step: impact of the environment

Adaptive motion control of an actor means that the environment has an impact on the actor motion and conversely. Informations about the environment and the actor must be available during the control process. The purpose of adaptive control motion is to decrease the amount of information entered into the computer by the animator. This is done by using existing informations about the scene and the actor. The system should also have an efficient representation of the geometry of the objects in order to automatically plan tasks as well as prevent collisions.

Girard [1987] gives a good example of this type of control applied to the motion of humans and animals on a flat terrain. At the low level, the animation is performed on a sequence of key positions of the limbs which define angle trajectories (direct kinematics) or Cartesian positions (inverse kinematics). These trajectories are calculated using optimizing criteria with kinematics or dynamics constraints.

The trajectory planning problem is classical and was extensively studied in robotics and Artificial Intelligence. For example, given the starting position of the actor hand and objects on a table, the problem is to find the trajectory to follow in order to avoid obstacles. For a synthetic actor, the problem is more complex due to the non-rigidity of the actor. We are currently work on the problem of walking without collision among obstacles. One strategy used is based on Lozano-Perez algorithm [Lozano-Perez and Wesley, 1979]. A 2D trajectory for the actor is obtained by projecting the octagonal cylinders surrounding the obstacles on the floor and constructing the visibility graph. Then, the obstacles are grown according to a selected projection of the actor. The result trajectory is then used as input to a positional constraint solver based on inverse kinematics.

7. Fourth step: task planning

Task planning is a major problem in robotics and artificial intelligence. The problem complexity is directly dependent on the generality of the actor micro world. Given a task description, the problem consists in decomposing the task in a sequence of elementary movements (see Fig.3).

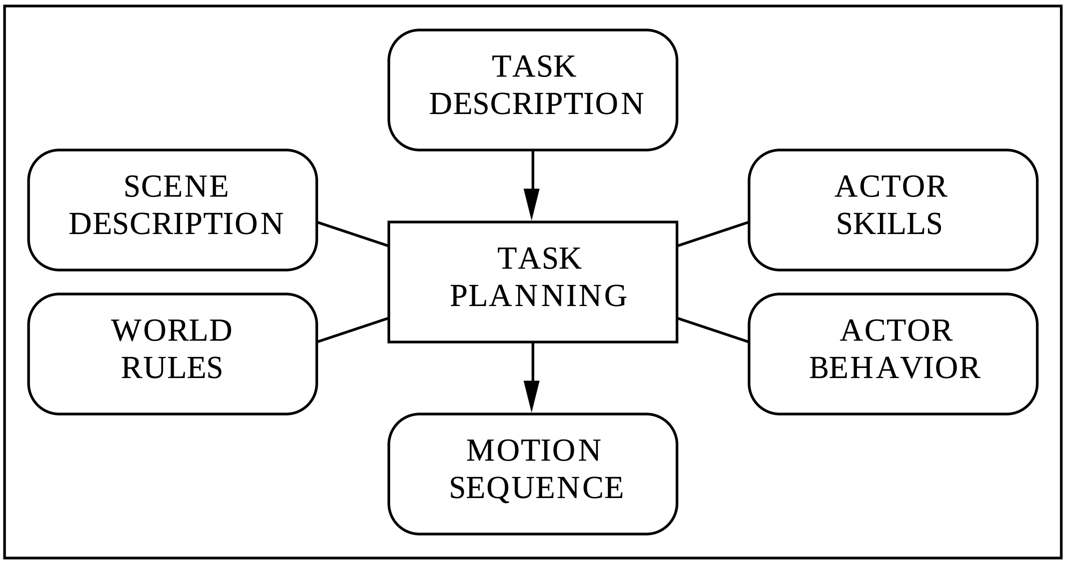

Fig.3 Task planning

In order to generate these movements, the system should possess the following informations:

. description of the scene (topology, position and orientation of the objects)
. database of the rules governing the micro world (e.g. it is necessary to stand up before walking)
. actor behavior (which shall modify the way of doing the movement; it corresponds to style parameters)
. library of elementary movements which may be done by the actor (actor skills)

For example, the task "answer to the phone" may be decomposed in the following sequence of elementary actions:

. stand up from his chair
. determine a trajectory that avoids obstacles in the actor motion
. walk according to the trajectory
. determine a trajectory that avoids obstacles for object grasping
. grasp the telephone
. answer

There are three ways of specifying tasks in a task-level system:

1. by example
2. by a sequence of model states
3. by a sequence of commands

The specification by example means for the operator "to perform the task at least once in order to explain it to the system." This is suitable in robotics, because the task may be physically specified by manually guiding the robot. This is of course impracticable in animation.

In the second type of method, the task is considered as a sequence of model states; each state is given by the configuration of all the objects in the environment. The configuration may be described by a set of spatial relationships. But what is the level of these relationships ? High-level relationships correspond for example to indicating that at a given time an object A must be at a certain height and in front of another object B. The problem in this case is that the set of relationships should be converted into a set of equations and inequations which may be very difficult to solve. Moreover, a set of configurations may overspecify a state. Low-level relationships may correspond to the coordinates of the objects at a certain time, which is a simple keyframe description. Several methods for obtaining configuration constraints from symbolic spatial relationships have been proposed [Popplestone et al., 1980; Taylor, 1976; Lozano-Perez, 1976].

The specification by a sequence of commands is the most suitable and popular. As stated by Zeltzer [1985], the animator can only specify the broad outlines of a particular movement and the animation system fills in the details. A non-expert user may be satisfied with the default movements, as generated by a task specification like WALK FROM A TO B. However, a high-end user may want nearly total control over every nuance of an actor's movement to make a sequence as expressive as possible. This means that the animator does need to access different levels of the control hierarchy in order to generate new motor skills and to tweak the existing skills.

Note that the transformation from a high level specification to a sequence of elementary motions is very similar to the problem of compiling. As in the processing of programming languages, three cases are possible: translation into a low-level code (classical compilers), translation into another programming language (preprocessor) and interpretation. In each case, the correspondence between the task specification and the motion to be generated is very complex. Consider three very essential tasks for a synthetic actor: grasping (Fig.4), walking (Fig.5) and talking (Fig.6).

Fig.4 Animation sequence: flower grasping

Fig.5 Walking

Fig.6 Talking

walking

To generate the motion corresponding to the task "WALK from A to B", it is necessary to take into account the possible obstacles, the nature of the terrain and then evaluate the trajectories which consist of a sequence positions, velocities and accelerations. Given such a trajectory, as well as the forces to be exerted at end effectors, it is possible to determine the torques to be exerted at the joints by inverse dynamics and finally the values of joint angles may be derived for any time. In summary, the task-level system should integrate the following elements: obstacle avoidance, locomotion on rough terrains, trajectory planning, kinematics and dynamics.

grasping

To generate the motion corresponding to the task "PICK UP the object A and PUT it on the object B", the planner must choose where to grasp A so that no collisions will result when grasping or moving them. Then grasp configurations should be chosen so that the grasped object is stable in the hand (or at least seems to be stable); moreover contact between the hand and the object should be as natural as possible. Once the object is grasped, the system should generate the motions that will achieve the desired goal of the operation. A free motion should be synthesized; during this motion the principal goal is to reach the destination without collision, which implies obstacle avoidance. In this complex process, joint evolution is determined by kinematics and dynamics equations. In summary, the task-level system should integrate the following elements: path planning, obstacle avoidance, stability and contact determination, kinematics and dynamics.

talking

To generate the motion corresponding to the task "SAY THE SENTENCE How are you? ", the system must analyze the sentence and separate it into phonemes, and then facial expressions corresponding to these phonemes must be selected. These expressions are themselves expressed as face deformations caused by muscles: jaw opening, eye opening, face folds etc. Once the expressions have been selected, the system should indicate to the computer at which times the expressions must be activated and generate the frames according to a law (spline for example). In summary, the task-level system should integrate the following elements: phonemes detection, selection of facial expression selections, handling of facial parameters, animation generation.

8. Fifth step: behavioral animation

Behavioral animation [Reynolds, 1987] corresponds to modeling the behavior of characters, from a path planning to complex emotional interactions between characters. The animator is responsible for the design of these behaviors; " his job is somewhat like that of a theatrical director: the character's performance is the indirect result of the director's instructions to the actor." Due to the personality of the character, his reactions may sometimes cause surprises. In an ideal implementation of a behavioral animation, it is almost impossible (as in a theatrical scene) to exactly play the same scene twice. You cannot walk exactly the same way from the same bar to home twice.

One current experiment is the design of an animation scene consisting of a small group of people walking together. The problem mainly consists of finding trajectories and using them as inputs to a positional constraint solver based on inverse kinematics.

The trajectories are obtained by a behavioral animation module. As in [Reynolds, 1987], positions, velocities and orientations of the actors are known from the system at any time. The animator may control several global parameters:

. weight of the obstacle avoidance component
. weight of the convergence to the goal
. weight of the centering of the group
. weight of the velocity equality
. maximum velocity
. maximum acceleration
. minimum distance between actors

References

Armstrong WW and Green MW (1985) Dynamics for Animation of Characters with Deformable Surfaces in: N.Magnenat-Thalmann and D.Thalmann (Eds) Computer-generated Images, Springer, pp.209-229.

Armstrong WW and Green MW (1985b) The Dynamics of Articulated Rigid Bodies for Purposes of Animation, The Visual Computer, Vol.1, No4, pp.231-240.

Arnaldi B., Dumont G., Hégron G., Magnenat-Thalmann N. and Thalmann D. (1989) Animation Control with Dynamics in: State-of-the-Art in Computer Animation, Springer, Tokyo (to appear)

Badler NI (1982) (Ed.) Special issue on "Modeling the Human Body for Animation", IEEE Computer Graphics and Applications, Vol.2, No9

Badler NI and Morris MA (1982) Modelling Flexible Articulated Objects, Proc. Computer Graphics '82, Online Conf., pp.305-314.

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

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.

Badler NI, Manoochehri KH and Walters G (1987) Articulated Figure Positioning by Multiple Constraints, IEEE Computer Graphics and Applications, Vol.7, No 6, pp.39-51

Boisvert D, Magnenat-Thalmann N, Thalmann D (1989) An Integrated View of Synthetic Actors, Proc. Computer Graphics International '89 (to appear)

Burtnyk N, Wein M (1971) Computer-generated Key-frame Animation, Journal of SMPTE, 80, pp.149-153.

Catmull E (1972) A System for Computed-generated movies, Proc. ACM Annual Conference, pp.422-431.

Forest L, Magnenat-Thalmann N, Thalmann D (1986) An Integration of Keyframe and algorithmic animation, Proc. Computer Graphics Tokyo '86.

Forsey D and Wilhelms J, Techniques for Interactive Manipulation of Articulated Bodies Using Dynamics Analysis, Proc. Graphics Interface '88, pp.8-15

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

Girard M and Maciejewski AA (1985), Computational Modeling for the Computer animation of Legged Figures, Proc. SIGGRAPH '85, pp.263-270

Hill DR, Pearce A and Wyvill B (1988) Animating Speech: an Automated Approach Using Speech Synthesised by Rules, The Visual Computer, Vol.3, No5

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

Korein JU, Badler NI, Techniques for Generating the Goal-directed Motion of Articulated Structures, IEEE CG&A, Nov.1982, pp.71-81

Kroyer B (1986) Animating with a Hierarchy, Seminar on Advanced Computer Animation SIGGRAPH '86.

Lewis JP, Parke FI (1987) Automated Lip-synch and Speech Synthesis for Character Animation, Proc. CHI '87 and Graphics Interface '87, Toronto, pp.143-147.

Lozano-Perez (1976) The Design of a Mechanical Assembly System, Artificial Intelligence Laboratory, MIT, AI TR 397

Lozano-Perez T and Wesley MA (1979) An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles, Comm.ACM, Vol.22, No10, pp. 560-570.

Magnenat-Thalmann and Thalmann D (1985) Computer Animation: Theory and Practice, Springer, Tokyo

Magnenat-Thalmann N, Laperrière R and Thalmann D (1988) Joint-dependent Local Deformations for Hand Animation and Object Grasping, Proc. Graphics Interface '88

Magnenat-Thalmann N, Primeau E, Thalmann D (1988b) Abstract Muscle Action Procedures for Human Face Animation, The Visual Computer, Vol.3, No5

Nahas M, Huitric H and Saintourens M (1988) Animation of a B-spline Figure, The Visual Computer, Vol.3, No5

Parke F.I. (1974) A Parametric Model for Human Faces, PhD dissertation, University of Utah, department of Computer Science

Pearce A, Wyvill B, Wyvill G and Hill D (1986) Speech and expression: a Computer Solution to Face Animation, Proc. Graphics Interface '86, pp.136-140.

Platt S, Badler N (1981) Animating Facial Expressions, Proc. SIGGRAPH '81, pp.245-252.

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

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

Taylor RH (1976) The Synthesis of Manipulator Control Programs from Task-Level Specifications, Artificial Intelligence Laboratory, Stanford University, AIM-282

Waters K (1987) A Muscle Model for Animating Three-Dimensional Facial Expression, Proc. SIGGRAPH '87, Vol.21, No4, pp.17-24.

Wilhelms J (1987) Towards Automatic Motion Control, IEEE Computer Graphics and Applications, Vol.7, No 4, pp.11-22

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.

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

Zeltzer D (1985) Towards an Integrated View of 3D Computer Animation, The Visual Computer, Vol.1, No4, pp.249-259.