

# An integrated control view of synthetic actors

Dominique Boisvert  
MIRALab, Université de Montreal, Canada

Nadia Magnenat-Thalmann  
Centre Universitaire d'Informatique  
Université de Genève, Suisse

Daniel Thalmann  
Laboratoire d'Infographie  
Ecole Polytechnique Fédérale de Lausanne, Suisse

## Abstract

This paper presents the global design of a system for synthetic actor animation. We discuss the concepts that must be part of an ideal animation system: actor creation, key framing, positional constraints, adaptive motion control, trajectory planning and task planning. A complete diagram of the components of such a system is presented together with the relations between these components.

**keywords:** synthetic actor, task planning, kinematics, dynamics, constraints

## The synthetic actor approach

Three-dimensional human modeling and animation have existed for more than 15 years<sup>1 2</sup>. 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<sup>3 4 5</sup>, on facial animation<sup>6 7 8 9 10 11 12 13</sup> and hand animation<sup>14 15 16</sup>. More recently, researchers have incorporated mechanical aspects factors into the animation of articulated bodies<sup>17 18 19 20 21 22</sup>. Also task level factors have been studied by several researchers<sup>23 24 25 26</sup>. Other authors have tried to animate more realistic characters, from an image synthesis point of view, but using primitive methods like rotoscoping or image-based keyframe animation. The synthetic 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 personalities, 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.

### **Traditional computer animation**

Most authors <sup>2 24 27 28 29</sup> distinguish between two types of three-dimensional computer animation: key-frame animation and algorithmic animation.

Keyframe animation consists of the automatic generation of intermediate frames, called inbetweens, based on a set of key-frames supplied by the animator. There are two fundamental approaches to keyframe animation:

1. The inbetweens are obtained by interpolating the keyframe images themselves. This technique is called **image-based keyframe animation**. This is an old technique, introduced by Burtnyk and Wein <sup>30</sup>.
2. A way of producing better images is to interpolate parameters of the model of the object itself. This technique is called **parametric keyframe animation** <sup>24 28 29</sup>. 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.

In algorithmic animation, motion is algorithmically described. Physical laws are applied to parameters of the human figures (e.g. joint angles).

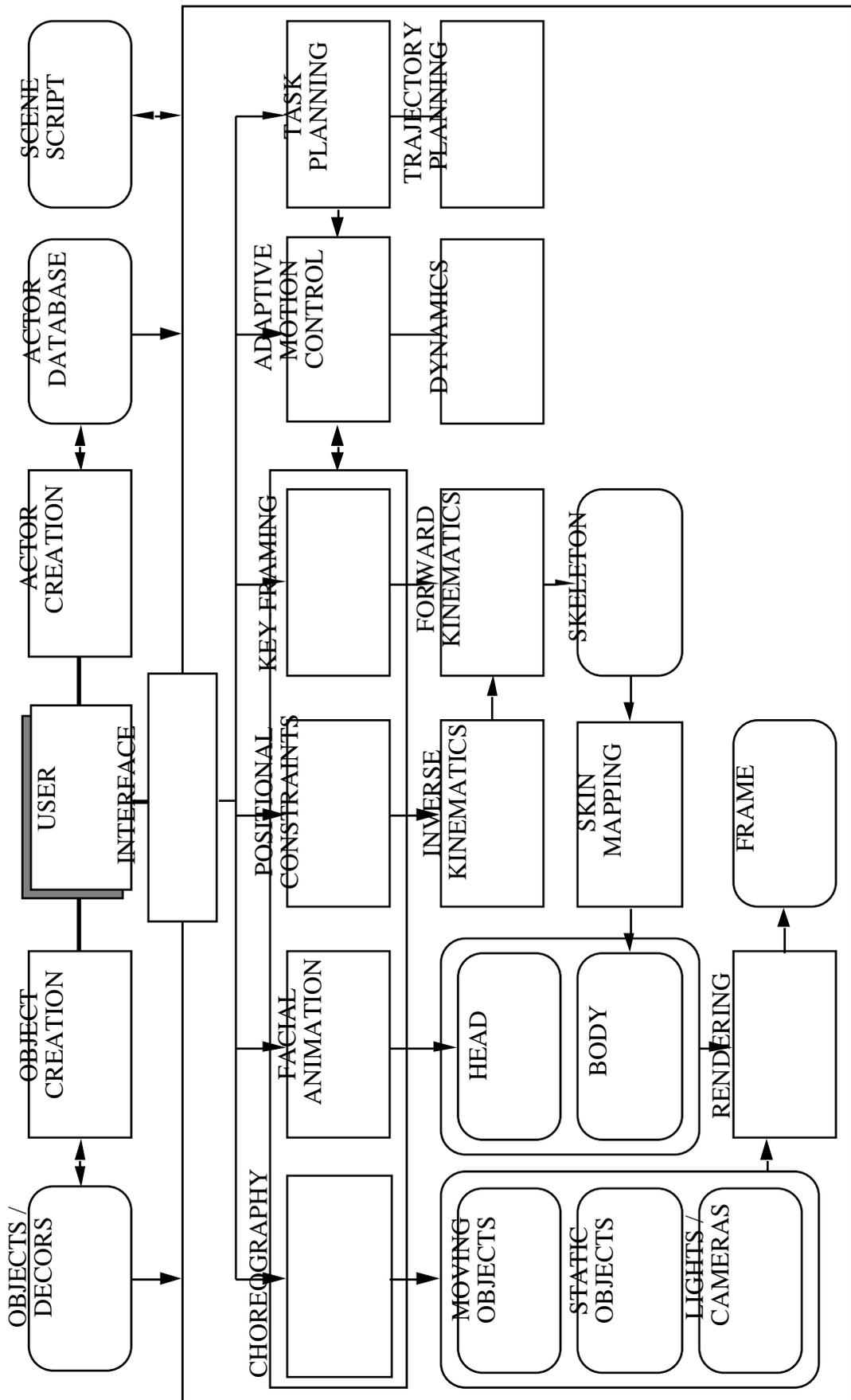


Fig.1 An integrated system

## Positional constraints

In this section, we address 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. This is a well-known problem in robotics, called the **inverse-kinematics problem**. It involves the determination of the joint variables from the given position and orientation of the end of the manipulator with respect to the reference coordinate system. The limb positioning is the key problem, because the independent variables in a robot as well as in a synthetic actor are joint variables. Unfortunately, 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 do exist. An example is given by manipulators with 6 joints, where the three joints nearest the end effector are all revolve and their axes intersect at one point (the wrist). Several solutions to the latter have been suggested in the context of animation<sup>31 32 33 34 35</sup>. Forsey and Wilhelms<sup>36</sup>, in their dynamics-based system, solve one constraint to a limb by linking it together with a pseudo-segment having a large mass value.

In order to make a synthetic actor sit down on a chair, 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.<sup>2</sup> 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.

The orientation of the hand or the foot should also be specified by the user. The orientation may be defined with respect to the limb, the actor or the world. For example, to ensure that the feet are flat on the floor, the user defines the foot orientation relatively to the world, whatever the position and orientation of the pelvis may be. The physical limitations of the joints have to be taken into account by the system. The interactivity of the system may be improved using input devices with multiple degrees of freedom.

A simple algorithm solving the positional constraint problem has been implemented in the Human Factory System. 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, . This is a well-known situation; as an example, when an actor is sitting down with a foot resting on the floor, he may rotate his knee around a hip-ankle axis. The variation of the rotation angle is constrained by the physical limitations of the joints. It may also be pointed out that the comfortable position for the knee (elbow) depends on the orientation of the foot (hand).

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.

### **Adaptive motion control**

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. Traditional animation techniques like rotoscopy or key framing cannot be considered as adaptive control techniques, because the animator has to explicitly control by hand the relation between the environment and the actors.

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 <sup>4</sup> 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 model should also produce realistic motion. In order to take into account internal and external forces acting on the actor, the system has to use a dynamics-based model. Techniques based on dynamics have already been used in computer animation <sup>17 18 19 20 21 22</sup>, 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.

To animate a body using kinematics (direct or inverse), the animator should specify the trajectories of the joint angles or the trajectories of the limb extremities. The trajectory must include time information (velocities and accelerations). Kochanek-Bartels<sup>37</sup> splines may be used for example; however, the realism of the motion is not guaranteed. A good compromise is an integration of both

models as suggested by Girard <sup>21</sup>, who obtained very good results using his system.

In the Human Factory system, we have decided to build a library of basic parametric motions (like walk and grasp) and each type of motion uses the kinematics when possible and only the dynamics when it is required for the realism of the motion. Each command produces a sequence of keyframes and positional constraints. A user which is not satisfied with the generated motion may then edit it.

## Trajectory planning

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. In the Human Factory system, any joint motion may be defined in three ways corresponding to the above classification:

1. Key values are selected at fixed times; spline interpolation is performed using the Kochanek-Bartels <sup>37</sup> approach.
2. The variation of angle is defined by a procedural law (parametric curve with evolution)
3. The trajectory may be generated using an automatic trajectory-planning algorithm. In our case, we use an anti-collision algorithm in a polytopic environment (n-dimension) with creation of a graph from hyperplane bundle. This method developed in our team seems more appropriate than the robotics methods <sup>38</sup>

## 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.2).

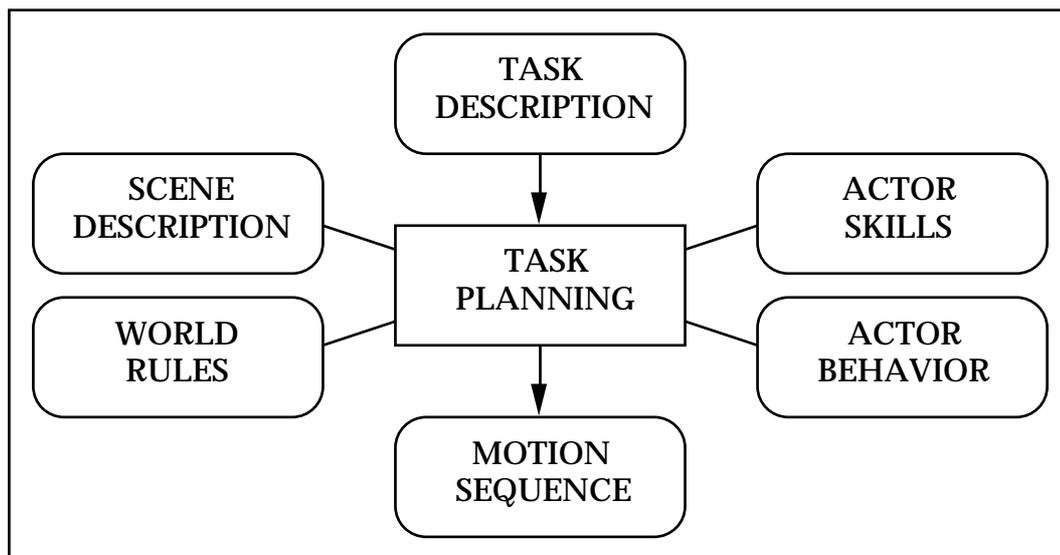


Fig.2 Task planning

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<sup>39 40 41</sup>.

The specification by a sequence of commands is the most suitable and popular. As stated by Zeltzer<sup>24</sup>, 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: walking, grasping and 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.

We have implemented in our animation system the Winograd<sup>42</sup> blocks world. Although this blocks world was created for natural language understanding, it is especially interesting for human animation, because it corresponds to planning motion sequences. To generate the motion sequence, we use a notation from compiling theory. This notation is similar to the syntactic diagrams used to define the syntax of a programming language. We assume terminal and non-terminal symbols. Terminal symbols in our case are generating commands: **GRASP B**, **MOVE B TO P**, **UNGRASP B**. Non-terminal symbols correspond to meta-commands such as: "places a block B1 at 3D location P on block B2" or "clears the top of the block B". We use the method of recursive descent popular in compiler construction. A procedure (which may be recursive) is written for each non-terminal symbol in order to construct a goal tree. Each node of the goal tree is a goal, which is only satisfied when their immediate subgoals are satisfied.

## References

- 
- 1 Badler NI (1982) (Ed.) Special issue on "Modeling the Human Body for Animation", IEEE Comp Graphics and Applications, Vol.2, No9
  - 2 Magnenat-Thalmann and Thalmann D (1985) Computer Animation: Theory and Practice, Springer, Toky
  - 3 Badler NI and Smoliar SW (1979) Digital Representation of Human Movement, ACM Computing Surv March issue, pp.19-38.
  - 4 Kroyer B (1986) Animating with a Hierarchy, Seminar on Advanced Computer Animation SIGGRAPH '86
  - 5 Forest L, Magnenat-Thalmann N, Thalmann D (1986) An Integration of Keyframe and algorithm animation, Proc. Computer Graphics Tokyo '86.
  - 6 Parke F.I. (1972) Animation of Faces, Proc. ACM Annual Conf., Vol.1

- 
- <sup>7</sup> Parke F.I. (1974) A Parametric Model for Human Faces, PhD dissertation, University of Utah, department Computer Science
- <sup>8</sup> Platt S, Badler N (1981) Animating Facial Expressions, Proc. SIGGRAPH '81, pp.245-252.
- <sup>9</sup> Pearce A, Wyvill B, Wyvill G and Hill D (1986) Speech and expression: a Computer Solution to I Animation, Proc. Graphics Interface '86, pp.136-140.
- <sup>10</sup> Hill DR, Pearce A and Wyvill B (1988) Animating Speech: an Automated Approach Using Spe Synthesised by Rules, The Visual Computer, Vol.3, No5
- <sup>11</sup> Lewis JP, Parke FI (1987) Automated Lip-synch and Speech Synthesis for Character Animation, Proc. '87 and Graphics Interface '87, Toronto, pp.143-147.
- <sup>12</sup> Magnenat-Thalmann N, Primeau E, Thalmann D (1988c) Abstract Muscle Action Procedures for Hu Face Animation, The Visual Computer, Vol.3, No5
- <sup>13</sup> Nahas M, Huitric H and Saintourens M (1988) Animation of a B-spline Figure, The Visual Computer, Vc No5
- <sup>14</sup> Catmull E (1972) A System for Computed-generated movies, Proc. ACM Annual Conference, pp.422-43
- <sup>15</sup> Badler NI and Morris MA (1982) Modelling Flexible Articulated Objects, Proc. Computer Graphics Online Conf., pp.305-314.
- <sup>16</sup> Magnenat-Thalmann N, Laperrière R and Thalmann D (1988b) Joint-dependent Local Deformations Hand Animation and Object Grasping, Proc. Graphics Interface '88
- <sup>17</sup> Armstrong WW and Green MW (1985) Dynamics for Animation of Characters with Deformable Surf: in: N.Magnenat-Thalmann and D.Thalmann (Eds) Computer-generated Images, Springer, pp.209-229.
- <sup>18</sup> Armstrong WW and Green M (1985b) The Dynamics of Articulated Rigid Bodies for Purpose Animation, The Visual Computer, Vol.1, No4, pp.231-240.
- <sup>19</sup> Wilhelms J and Barsky B (1985) Using Dynamic Analysis to Animate Articulated Bodies such as Hun and Robots, in: N.Magnenat-Thalmann and D.Thalmann (Eds) Computer-generated Images, Sprin pp.209-229.
- <sup>20</sup> Wilhelms J (1987) Towards Automatic Motion Control, IEEE Computer Graphics and Applications, Vc No 4, pp.11-22
- <sup>21</sup> Girard M (1987) Interactive Design of 3D Computer-animated Legged Animal Motion, IEEE Comp Graphics and Applications, Vol.7, No 6, pp.39-51
- <sup>22</sup> Isaacs PM and Cohen MF (1987) Controlling Dynamic Simulation with Kinematic Constraints, Bah Functions and Inverse Dynamics, Proc. SIGGRAPH'87, Computer Graphics, Vol.21, No4, pp.215-224
- <sup>23</sup> Zeltzer D (1982) Motor Control Techniques for Figure Animation, IEEE Computer Graphics Applications, Vol.2, No9, pp.53-59.
- <sup>24</sup> Zeltzer D (1985) Towards an Integrated View of 3D Computer Animation, The Visual Computer, Vc No4, pp.249-259.
- <sup>25</sup> Badler NI, Korein JD, Korein JU, Radack GM and Brotman LS (1985) Positioning and Animating Figure a Task-oriented Environment, The Visual Computer, Vol.1, No4, pp.212-220.
- <sup>26</sup> Badler NI, Manoochehri KH and Walters G (1987) Articulated Figure Positioning by Multiple Constrai IEEE Computer Graphics and Applications, Vol.7, No 6, pp.39-51
- <sup>27</sup> Hanrahan P and Sturman D (1985) Interactive Animation of Parametric Models, The Visual Comp Vol.1, No4, pp.260-266.
- <sup>28</sup> Steketee SN, Badler NI (1985) Parametric Keyframe Interpolation Incorporating Kinetic Adjustment Phrasing Control, Proc. SIGGRAPH '85, pp. 255-262.
- <sup>29</sup> Parke FI (1982) Parameterized Models for Facial Animation, IEEE Computer Graphics and Applicati Vol.2, No 9, pp.61-68
- <sup>30</sup> Burtnyk N, Wein M (1971) Computer-generated Key-frame Animation, Journal of SMPTE, 80, pp.149-1
- <sup>31</sup> Badler NI, Korein JD, Korein JU, Radack GM and Brotman LS, Positioning and Animating Human Fig

- 
- in a Task-Oriented Environment, *The Visual Computer*, Dec. 1985, pp.212-220.
- <sup>32</sup> Badler NI, Manoochehri KH and Walters G, Articulated Figure Positioning by Multiple Constraints, *IEEE CG&A*, June 1987, pp.28-38
- <sup>33</sup> Girard M and Maciejewski AA, Computational Modeling for the Computer animation of Legged Figure, *Proc. SIGGRAPH '85*, pp.263-270
- <sup>34</sup> Girard M, Interactive Design of 3D Computer-Animated Legged Animal Motion, *IEEE CG&A*, June 1987, pp.39-51
- <sup>35</sup> Korein JU, badler NI, Techniques for Generating the Goal-directed Motion of Articulated Structures, *IEEE CG&A*, Nov.1982, pp.71-81
- <sup>36</sup> Forsey D and Wilhelms J, techniques for Interactive Manipulation of Articulated Bodies Using Dynamic Analysis, *Proc. Graphics Interface '88*, pp.8-15
- <sup>37</sup> Kochanek D and Bartels R (1984) Interpolating Splines with Local Tension, Continuity and Bias Tension, *Proc. SIGGRAPH '84*, pp.33-41.
- <sup>38</sup> Lozano-Perez (1982) Task Planning in: Brady M (Ed.) *Robot Motion: Planning and Control*, MIT Press, Cambridge, Mass.
- <sup>39</sup> Popplestone RJ, Ambler AP and Bellos IM (1980) An Interpreter for a Language for Describing Assemblies, *Artificial Intelligence*, Vol.14, pp.79-107
- <sup>40</sup> Taylor RH (1976) The Synthesis of Manipulator Control Programs from Task-Level Specifications, *Artificial Intelligence Laboratory, Stanford University*, AIM-282
- <sup>41</sup> Lozano-Perez (1976) The Design of a Mechanical Assembly System, *Artificial Intelligence Laboratory MIT*, AI TR 397
- <sup>42</sup> Winograd T (1972) *Understanding Natural Language*, Academic Press, NY