Chapter 1

Human Modelling and Animation

Daniel Thalmann

This report brings attendees up-to-date on the frontiers of research in human modeling, animation, and rendering. Research in this area implies the development of numerous techniques: improving the physical aspects (shapes, colors, textures) of the actors, improving the deformation of limbs during motion, improving facial expressions and deformations, specifying tasks to be performed using natural language, and simulating behaviors. The first problems we discuss are those of shape creation and animation. For shape creation, we show the impact of new 3D devices and Virtual Reality techniques on the design of the human body and face. We briefly review the last improvements in basic techniques of motion control of articulated bodies. We then discuss the problem of modeling deformations of human bodies during animation. We also present the three main types of facial animation systems classified in terms of driving mechanism or animation control. We also explore recent methods of improving the realism of human appearance in computer-generated films. We present methods for designing and animating clothes using deformable models involving physical principles that govern rigid and nonrigid dynamics,. We review techniques for rendering fur and hair and modeling hairstyles. We then emphasize recent developments and problems in task-level animation and behavioral animation. In particular we present last developments in walking models and grasping techniques. Finally, we introduce the general concept of autonomous actors reacting to their environment and making decisions based on perception systems, memory and reasoning. With such an approach, we should be able to create simulations of actors moving in a complex environment they "know" and "recognize", or actors playing ball games based on their perceptions of sight and touch. Examples of vision-based animation of synthetic actors will be presented.

1.0 Virtual Humans

1.1 Introduction

The ultimate reason for developing realistic-looking virtual humans is to be able to use them in virtually any scene that recreates the real world. However, a virtual scene -- beautiful though it may be -- is not complete without people.... Virtual people, that is. Scenes involving synthetic actors imply many complex problems we have been solving for several years [1]. A *synthetic actor* is defined as a human-like autonomous actor completely generated by computer. Applications of synthetic actors are unlimited: in the near future, any human being, dead or alive, may be recreated and placed in any new situation, with no dependence on any live action model. Digital scene simulation will be possible for landscapes with human beings, cinema or theater actors, and spatial vessels with humans; any human behaviour may be simulated in various situations, scientific as well as artistic. From a user point-of-view, TV announcers may be simulated, or people may walk inside their dream house before the house is built. In the biomedical world, applications are also numerous: deformations of the back and the impact of bad postures, simulation of language dysfunctions and visual dysfunctions. Even in sports education, good positions may be shown as well as the effect of acting muscles. Human beings in dangerous situations may be also simulated: car accidents, airplane crashes, fires, explosions, etc.

Human-like synthetic actors have very irregular shapes that are hard to construct, especially for well-known personalities where a recognition factor comes into play. Once an initial human shape has been created, though, this shape should adapt and change during the animation. Ensuring the continuity and realism of the deformed surfaces is a very complex problem. The process of animating humans is very involved and should be split into two areas: body motion control and facial animation. Basically a synthetic actor is structured as an articulated body defined by a skeleton. Skeletal animation consists of animating joint angles. There are two main ways to do this: parametric keyframe animation and physics-based animation. Another complex objective is modeling human facial anatomy exactly, including movements to satisfy both structural and functional aspects of simulation.

1.2 Background

The first computerized human models were created 20 years ago by aeroplane and car manufacturers. The main idea was to simulate a very simple articulated structure for studying problems of ergonomics. In the seventies, researchers developed methods to animate human skeletons, mainly based on interpolation techniques. Bodies were represented by very primitive surfaces like cylinders, ellipsoids or spheres. At the same time, the first experimental facial animation sequences appear. The *Juggler* (1982) from the former Information International Inc. (III) was the first realistic human character in computer animation; the results were very impressive; however, the human shape was completely digitized, the body motion had been recorded using 3D rotoscopy and there was no facial animation. In the mid-eighties, researchers started to use the laws of physics. Based on dynamic simulations, it was possible to generate complex motions with a great deal of realism. Even an ordinary human activity like walking is however too complex to be simulated by the laws of dynamics alone. Currently, dynamic simulation is used to portray a few motions of robots or worms as in the film *The Audition* from Apple Computer [2]. Moreover, dynamic simulation always generates the same motions, unrealistic behavior for humans. Two people, even with the same physical characteristics, do not move in the same way. Even one individual does not move in the same way all the time. A behavioral approach to human animation will be eventually necessary to lend a certain credibility to such simulations.

2.0 Creation of Human Shapes

The human body has a complex and irregular surface that is difficult to model. A surface like a human face is also irregular and composed of bossels and depressions. From a geometric modelling point-of-view, the most popular model for human body and face is still the polygonal mesh. Although often expensive in terms of CPU time, polygonal models of 3D objects are the most common ones. One of the reasons for using this kind of model is that the latest superworkstations provide facilities to process them by hardware (fast rendering and matrix operations).

2.1 Digitizing Methods

Traditional methods to create irregular shapes like human shapes are based on digitizing. Three popular methods have been extensively used:

- *Three-dimensional reconstruction from two-dimensional photographs.* Two or three projections (photos) are entered and the computer is used to derive the 3D coordinates. Synthetic Marilyn and Bogey in the film *Rendez-vous in Montreal* [3] were created using this approach.

- *Reconstruction from cross sections.* This popular method consists of reconstructing an object from a set of serial cross sections, like tracing the contours from a topographic map. This method [4] has been used to create *Eglantine*, a computerized mannequin, who never existed before.

- *Three-dimensional digitizing.* The technique is simply to enter the 3D coordinates using a 3D digitizer. We used for example, the Polhemus 3D-digitizer (based on magnetic fields) to create various objects. The method is less time-consuming than the two other methods, because no photos are needed. However, there are limitations in the shapes, which can be digitized; cavities and small parts cannot be entered.

2.2 Shape Interpolation Between Human Faces

A shape interpolation [4] consists in generating an inbetween human face from two given human faces. The main problem of this method is that both original faces may have different numbers of facets and vertices. The technique consists in extracting profiles of both human faces from selected planes and generating two grids which correspond to the original faces. Then a correspondence is establishing between the profiles, then the correspondence between the parallel sections is found using a similar method. Now the correspondence between points is straightforward. Finally an inbetween human face is just obtained by linear interpolation. The technique has been succesfully used in the film *Galaxy Sweetheart* to transform th synthetic actress Marilyn into the synthetic actor Bogey.

2.3 Local Deformations and Sculpting Tools

For creating human shapes, local deformations are probably the best for the modification and even creation of human surfaces. Several authors have proposed methods to perform limited deformations. In 1984, Barr [5] propose global deformations. In 1986, Sederberg and Parry [6] introduce Free Form deformations extended by Coquillart [7] in 1990. Numerous methods based on parametric surfaces are extensively used in CAD and CAE. As these methods deal with control points, they are not suitable for human modeling except when no resemblance with existing people is required [8 9]. Second, field functions [10] model free-form surfaces and their local deformations, but the application of this for the creation of well-known personalities seems difficult.

Allan et al. [11] proposed a general method for manipulating polygonal meshes. They introduced a basic operation "move-vertex", which specifies a new 3D position for a specific vertex called the current vertex. Their basic operation is extended in several ways: definition of a range of influence around the current vertex, decay function over the range of influence, binding and anchoring. Other operations include stretch, grow and randomize.

A realistic human character may be produced with a method similar to the modelling in clay, work which essentially consists of adding or eliminating bits of material, and turning the object around the object when the shape has been set up. An elegant solution is the use of a sculpting software [12] based on the Ball and Mouse metaphor [13]. The metaphor is based on an interactive 3D input device like the Spaceball, a 6 degree-of-freedom device. When used in conjunction with a common 2-D mouse, the Spaceball being held in one hand and the mouse in the other, full three-dimensional user interaction is achieved. The Spaceball device is used to move the object being sculpted around in order to examine it from various points of view, while the mouse carries out the picking and deformation work on a magnified image so that every small detail may be seen in real time. In this way, the user not only sees the object from every angle but can also apply and correct deformations from every angle interactively.

3.0 Motion Control Methods of Articulated Bodies

3.1 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** [14] 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. The flexing is a rotation of the limb which is influenced by the joint and cause the motion of all limbs linked to this joint. This flexing is made relatively to the joint point and a flexing axis which has to be defined. The **pivot** makes rotate the flexing axis around the limb which is influenced by the joint. The **twisting** causes a torsion of the limb which is influenced by the joint. The direction of the twisting axis is found similarly to the direction of the pivot. Fig. 1 shows an example of skeleton for a human-like figure.

3.2 Kinematics methods for skeleton animation

3.2.1 Rotoscopy and keyframe animation

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. As already mentioned, 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.

Name	Number	Angles
VERTEBRA 1	2	FTP
VERTEBRA 2	3	FTP
VERTEBRA 3	4	FTP
VERTEBRA 4	5	FTP
VERTEBRA 5	6	FTP
LEFT CLAVICLE	7	FP
RIGHT CLAVICLE	11	FP
LEFT SHOULDER	8	FTP
RIGHT SHOULDER	12	FTP
LEFT ELBOW	9	FT
RIGHT ELBOW	13	FT
LEFT WRIST	10	FP
RIGHT WRIST	14	FP
LEFT HIP	15	F
RIGHT HIP	20	F
LEFT THIGH	16	FTP
RIGHT THIGH	21	FTP
LEFT KNEE	17	F
RIGHT KNEE	22	F
LEFT ANKLE	18	F
RIGHT ANKLE	23	F
LEFT TOE	19	F
RIGHT TOE	24	F

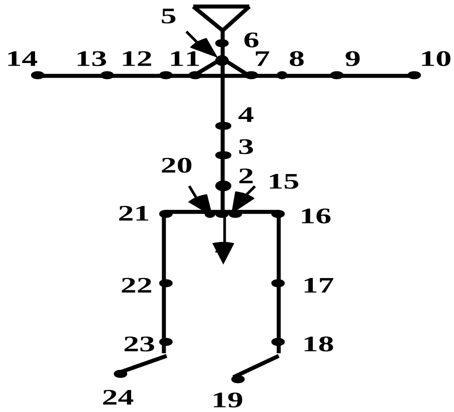

FIGURE 1. A basic skeleton with the joints angles (F: flexion, T: twisting, P:pivot)

3.2.2 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. Animating articulated limbs by interpolating key joint angles corresponds to forward kinematics.

The use of *inverse-kinematics* [[15]] 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 [[16] [17] [18] [19]]. For generating goal-directed movements such as moving the hand to grasp an object, it is necessary to compute inverse kinematics. Fig. 2 shows the principles of forward and inverse kinematics. In a typical 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.

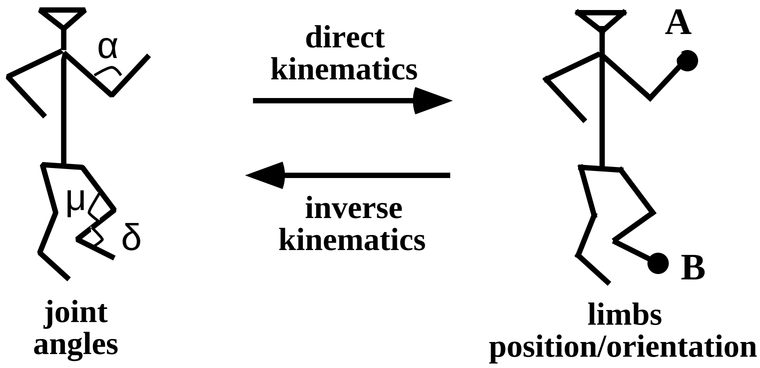

FIGURE 2. Forward and inverse kinematics

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. [20] 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.

3.3. Dynamics

3.3.1 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. These forces and torques may be of various kinds:

- torques coming from parent and child links,
- forces at the hinges,
- external effects such as contact with objects or arm-twisting.

There are three advantages of introducing dynamics into animation control [21]:

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

There are also serious disadvantages.

- Parameters (e.g. forces or torques) are sometimes very difficult to adjust, because they are not intuitive.
- The amount of CPU time required to solve the motion equations of a complex articulated body using numerical methods.
- Dynamics-based motions are too regular.

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.

In dynamic-based simulation, there are also two problems to be considered: the *forward dynamics* problem and the *inverse-dynamics* problem. The forward dynamics problem consists of finding the trajectories of some point (e.g. an end effector in an articulated figure) with regard to the forces and torques that cause the motion. The inverse-dynamics problem is much more useful and may be stated as follows: determine the forces and torques required to produce a prescribed motion in a system. For an articulated figure, 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.

3.3.2 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 [22]. The *Newton-Euler formulation* [23] 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. [24] use a recursive method to design a near real-time dynamics algorithm and implement it in a prototype animation system. Wilhelms and Barsky [25] use the **Gibbs–Appel formulation** for their animation system Deva; however, the cost of solving for accelerations is prohibively 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 [26] use the D'Alembert formulation in their DYNAMO system. Also, Arnaldi et al. [21] 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.

3.3.3 Constraint-based Methods

Isaacs and Cohen [27] 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. [28] using *energy constraints.* Using *dynamic constraints*, Barzel and Barr [29] 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 [30] extend dynamic constraints to flexible models using reaction constraints and optimization constraints. Witkin and Kass [31] 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.

4.0. Surface body animation and deformation

Once the motion of the skeleton is designed, the realism of motion needs to be improved not only from the joint point-of-view, but also in relation to the deformations of bodies during animation. For animating rigid bodies like robots, a simple mapping between the surface and the skeleton is needed. For living beings, the surfaces should be transformed according to the wire-frame model ensuring an automatic continuity between the different surfaces and automatic deformations during the animation process. Deformations of animated objects have raised a lot of interests in the recent past years. Many deformations models have been proposed but less of them have been applied in the field of human body deformations. The complexity of the constitution of the body makes things really difficult in the sense that so many different materials from bones to muscles to fat tissues come in play that there is no homogeneous behavior.

Magnenat-Thalmann et al [[32]] have proposed a model of animation for the hand called JLD deformations, joint dependent local deformations, where deformations are handled by geometrical operators adapted to the joint of the skeleton they are applied to. The case of the hand is especially complex, as deformations are very important when the fingers are bent, and the shape of the palm is very flexible. Links of fingers are independent and the JLD operators are calculated using a unique link-dependent reference system. For the palm, JLD operators use reference systems of several links to calculate surface mapping. In order to make the fingers realistic, two effects are simulated: joint roundings and muscle inflation. The hand mapping calculations are based on normals to each proximal joint.

Gourret et al [[33]] have presented a model simulating hand deformations and interactions with a representation using finite element methods (see Fig.3). Chadwick et al. [[34]] built an animation system for cartoon characters consisting of three layers : skeleton, muscles and skin where muscles are represented as control base for free form deformations deformed according to skeleton movements and applied to the skin. Gascuel et al. [[35]] have defined a structure where control points of a spline based skin are represented as endpoint of springs bound to the skeleton. Chen and Zeltzer [[36]] have provided a model of muscle based on finite element method. Delingette et al. [[37]] have defined a model for object representation based on a simplex mesh.The accuracy of this model for physically based animation is used to perform 3D morphing,extraction from range images and animating a hand. Turner et al [[38]] proposed an elastic surface layer model where the different layer are the skeleton, bones, muscles, fat tissues and skin. Skin is modelled as an elastic surface, fat tissues are defined as a thickeness between muscles and skin, muscles are modelled using geometrical solid surfaces.

5.0 Facial animation

The face is a small part of a human, but it plays an essential role in the communication. People look at faces for clues to emotions or even to read lips. It is a particular challenge to imitate these few details. An ultimate objective therefore is to model human facial anatomy exactly including its movements to satisfy both structural and functional aspects of simulation. However, this involves many problems to be solved concurrently. The human face is a very irregular structure, which varies from person to person. The problem is further compounded with its interior details such as muscles, bones and tissues, and the motion which involves complex interactions and deformations of different facial features.

FIGURE 3. Ball deformed by the hand

5.1 The Basic Facial Animation Models

The complexity of facial models leads to what is commonly called facial expressions. The properties of these facial expressions have been studied for 25 years by Psychologist Ekman, who proposed a parameterization of muscles with their relationships to emotions: the *Facial Action Coding System* (FACS) [39]. FACS describes the set of all possible basic actions performable on a human face. There has been extensive research done on basic facial animation and several models have been proposed. In the early models proposed by Parke [40 41], he has used a combination of digitized expressions and linear interpolation of features such as eyelids and eyebrows and rotations for jaw. The set of facial parameters is based on observation and the underlying structures that cause facial expression. The face is modeled as a collection of polygons manipulated through the set of parameters which may be interpolated.

Platt and Badler [42] have presented a model of a human face and developed a notational system to encode actions performable on a face. The notation drives a model of the underlying muscle structure which in turn determines the facial expression. The model simulates points on the skin, muscle, and bone by a set of interconnected 3D network of points using arcs between selected points to signify relations. Waters [43] proposes a muscle model which is not specific to facial topology and is more general for modifying the primary facial expression. In this model, muscles are geometric deformation operators which the user places on the face in order to simulate the contraction of the real muscles. Two types of muscles are created linear/parallel muscles that pull and sphincter muscles that squeeze. These muscles are independent of the underlying bone structure, which makes the muscle model independent of specific face topology. The control parameters are based on FACS. Magnenat Thalmann et al. [44] introduced a way of controlling the human face based on the concept of *abstract muscle action* (AMA) procedures. An AMA procedure is a specialized procedure which simulates the specific action of a face muscle. AMA procedures work on certain regions of the human face which must be defined when the face is constructed. Each AMA procedure is responsible for a facial parameter corresponding approximately to a muscle, for example, vertical jaw, close upper lip, close lower lip, lip raiser etc. A facial expression is considered as a group of facial parameter values obtained by the AMA procedures in different ways. Fig.4 shows an example of facial expression.

FIGURE 4. Facial expression

Terzopoulos and Waters [45] have extended the Waters model, using three layered deformable lattice structures for facial tissues. The three layers correspond to the skin, the subcutaneous fatty tissue, and the muscles. The bottom surface of the muscle layer is attached to the underlying bone. The model uses physically-based technique. Parke [46] reviews different parameterization mechanism used in different previously proposed models and introduces the future guidelines for ideal control parameterization and interface. Ideal parameterization is in fact a universal parameterization which would enable all possible individual faces with all possible expressions and expression transitions. Recently several authors have provided new facial animation techniques based on the information derived from human performances [47 48 49]. The information extracted is used for controlling the facial animation. These performance driven techniques provide a very realistic rendering and motion of the face. Kurihara and Arai [50] introduced a new transformation method for modeling and animating the face using photographs of an individual face. The transformation method enables the movement of points in the skin mesh to be determined by the movement of some selected control points. Texture mapping is used to render the final image. Kalra et al. [51] propose the simulation of muscle actions based on *Rational Free Form Deformations* (RFFDs) an extension of Free form deformation (FFD) [6].

5.2 Speech, Emotion and Synchronization

Most of the facial movements result from either speech or the display of emotions; each of these has its own complexity. However, both speech and emotions need a higher level specification of the controlling parameters. The second level parameterization used in speech animation is usually in terms of phonemes. A phoneme is a particular position of the mouth during a sound emission. These phonemes in turn control the lower level parameters for the actual deformations. Similarly, emotion is a sequence of expressions, and each expression is a particular position of the face at a given time. In order to have a natural manipulation of the speech and emotions there is a need of some synchronization mechanism. Efforts for *lip synchronization* and to automate the speech initiated with the first study of Lewis and Parke [52]. In their approach, the desired speech is spoken and recorded, the recording is then sampled and analyzed to produce a timed sequence of pauses and phonemes. Hill et al. [53] have introduced an automatic approach to animate speech using speech synthesized by rules. Magnenat-Thalmann et al. [44] have used lip synchronization based on AMA procedures. Let us have an example: For the phoneme "I" (as in "it"), the teeth are slightly open and the commissures are horizontally pulled towards the outside (risorius muscle); this corresponds to 10% of the AMA procedure VERTICAL_JAW, 50% of the AMA procedure LEFT_RISORIUS and 50% of the AMA procedure RIGHT_RISORIUS. At script level, a script in facial animation is a collection of multiple tracks, where each track is a chronological sequence of keyframes for a given facial parameter. On each track, a percentage of a facial parameter or the facial expression may be fixed for a given time. For example, "KID" will be pronounced by a character, indicating that the phoneme "K" is used at a given time, the phoneme "I" a short time later, then the phoneme "D". Then the software will progressively transform the facial expression corresponding to the phoneme "K" in order to obtain the facial expression corresponding to the phoneme "I", then to the phoneme "D".

In another system, Kalra et al. [54] introduce a *multi-layer approach* where, at each level, the degree of abstraction increases. The high level layers are the most abstract and specify "what to do", the low level layers describe "how to do". This results in a system where complexity is relatively low from the point of view of the animator. The defined entities correspond to intuitive concepts such as phonemes, expressions, words, emotions, sentences and eye motion, which make them natural to manipulate. A manipulation language, HLSS, is provided to ensure synchronization while manipulating these entities.

6.0 Clothes, skin, hair and beards

6.1 Cloth Modelling and Animation

Cloth animation in the context of human animation involves the modeling of garments on the human body and their animation. In the film "Rendez-vous à Montréal" [3] featuring Humphrey Bogart and Marilyn Monroe, clothes were simulated as a part of the body with no autonomous motion (see Figure 5). For modeling more realistic clothes, two separate problems have to be solved: cloth animation without considering collisions (only the basic shape and the deformations due to gravity and wind are considered), and collision detection of the cloth with the body and with itself.

In a geometric approach, the shape of flexible objects is entirely described by mathematical functions. It is not very realistic and cannot create complicated deformable clothes, but it is fast. The geometric approach is suitable for representing single pieces of the objects or clothes with simple shapes, which are easily computed, but geometric flexible models like Weil's model [55] or Hinds and McCartney's model [56] have not incorporated concepts of quantities varying with time, and are weak in representing physical properties of cloth such as elasticity, anisotropy, and viscoelasticity. Only physical models like Terzopoulos' model [57] and Aono's model [58] may correctly simulate these properties. Another interesting approach by Kunii and Gotoda [59] incorporates both the kinetic and geometric properties for generating garment wrinkles. Lafleur et al. [60] addresses the problem of detecting collisions of very flexible objects, such as clothes, with almost rigid bodies, such as human bodies. In their method, collision avoidance also consists of creating a very thin force field around the obstacle surface to avoid collisions. This force field acts like a shield rejecting the points. The volume is divided into small contiguous non-overlapped cells which completely surround the surface. As soon as a point enters into a cell, a repulsive force is applied. The direction and the magnitude of this force are dependent on the velocities, the normals and the distance between the point and the surface.

FIGURE 5. False clothes (from Rendez-vous à Montréal, 1987)

More recently, Carignan et al. [61] discuss the use of physics-based models for animating clothes on synthetic actors in motion. In their approach, cloth pieces are first designed with polygonal panels in two dimensions, and are then seamed and attached to the actor's body in three dimensions [62]. They describe a new approach to the problem of handling collisions among the cloth elements themselves, or between a cloth element and a rigid object like the human body. In their approach, they work as a tailor does, designing garments from individual two-dimensional panels seamed together. The resulting garments are worn by and attached to the synthetic actors. When the actors are moving or walking in a physical environment, cloth animation is performed with the internal elastic force and the external forces of gravity, wind, and collision response. Carignan et al. choose Terzopoulos' elastic surface model [57] for their system with the damping term replaced by a more accurate. The fundamental equation of motion corresponds to an equilibrium between internal forces (Newton term, resistance to stretching, dissipative force, resistance to bending) and external forces (collision forces, gravity, seaming and attaching forces, wind force). To apply the elastic deformable surface model, the polygonal panel should be discretized using the finite difference approximation method. Carignan et al. propose a new algorithm to calculate the elastic force on an arbitrary element. This algorithm is effective for discretizing not only an arbitrary polygonal panel (concave or convex), but also other kinds of polygonal panels with holes inside them. The constraints that join different panels together and attach them to other objects are very important. Two kinds of dynamic constraints [29] are used during two different stages. When the deformable panels are separated, forces are applied to the elements in the panels to join them according to the seaming information. The same method is used to attach the elements of deformable objects to other rigid objects. When panels are seamed or attached, a second kind of constraint is applied which keeps a panel's sides together or fixed on objects. Fig. 6 and Color Plate 1 show examples of a dressed synthetic actress.

FIGURE 6. Synthetic actress with clothes

6.2 Skin

The facial images arising from conventional type of rendering (e.g. continuous polygon shading) tend to look rather artificial and cartoon-like. This is particularly disconcerting for faces that are meant to be human in nature. All physical surfaces have a detailed structure visible to the human eye; this provides a tremendous amount of information about the nature of a substance. A means is required to communicate whether skin is smooth or rough, its color, reflectance and pigmentation.As synthetic skin texture are very complex to generate, texture mapping is the most popular way. Use of texture mapping for realism is not very recent [63 64 65]. However, most of the applications deal with mapping an image on parametric surfaces and solid modeling primitives. Consequently, these techniques in general include mapping a full image onto a surface. For facial image synthesis, merely projecting (mapping) a full image on the subject's face is not enough. There is a need to establish the positional certainty for some parts of the model with the source image to be mapped. This is essential particularly when the 3D face model is not constructed from the image(s). Recently, efforts have been made to apply texture-mapping technique to enhance rendering of faces. Some of the techniques [66] employ registered texture mapping which although give realistic rendering, involve enormous data-handling. In addition, it requires special hardware for registering the data. Waters and Terzopoulos [67] used an adaptive meshing technique which can be used to design coarser, non-uniform meshes capturing the essential structure of the high resolution facial maps from laser scanners. The technique however, needs a special hardware and is time consuming. The method used by Kurihara et al. [68] uses multiple photographs to reconstruct the 3D model of a face using interpolation in 2D cylindrical space. The photographs are composed into one picture using a weight function for each photograph in 2D cylindrical space. In another approach by Yau et al. [69] pre-stored sub-images are manipulated for animation, which is not very convenient and user-friendly. Kalra and Magnenat Thalmann [70] also propose a technique based on texture mapping of photos of real faces. A separate tool for matching the 3D facial topology on a given picture/photo of a face is developed. Only a few feature points are selected from the 3D model to exactly match the corresponding points on the picture. Delaunay triangulation is used to connect these points. These points can be moved and displaced on the picture interactively. An interpolation scheme in a triangular domain is used to get the desired texture coordinates.

6.3 Hair and Beards

In the field of human animation, hair presents perhaps the most challenging rendering problem and therefore has been one of the least satisfactory aspects of human images rendered to date. The difficulties of rendering hair result from the large number and detailed geometry of the individual hairs, the complex interaction of light and shadow among the hairs, and the small scale of the hair width in comparison with the rendered image. The rendering of hair therefore constitutes a considerable anti-aliasing problem in which many individual hairs, reflecting light and casting shadows on each other, contribute to the shading of each pixel. Creating fur and hair styles even in artificial beards, moustaches and furs is a complex and meticulous task involving the treatment of numerous segments of hair, their placement on the surface, form, material features and their dynamic aspects.

The first works in synthetic hairs were conceived to render fur-like volumes [71 72 73] and furry animals [74]. Concerning the synthesis of human hair, some recent models were developed to create and animate hair styles. Watanabe and Suenaga [75] presented a trigonal prism model of human hair. Each individual hair is a series of short trigonal prisms. Several kinds of hair can be created changing the parameters of length, direction vector, thickness, twist angle, and number of trigonal prisms. In order to facilitate the design work, hairs are grouped in wisps. The rendering method is based on sellecting areas for the backlighting effect and using a doble z-buffer.The animation is determined by defining a parabola trajectory of wisps in order to approximate their location according to the initial velocities and acceleration. As an example of their model, they present various images of hair styles with varied wisp parameters, as well as a sequence of hair animation. Rosenblum, Carlson and Tripp III [76] developed a technique to simulate human hair structure and dynamics. The hair structure is defined by connected straight cylindrical segments with a fixed width and it is modeled as a linear series of masses, springs and hinges. Hairs are rendered using z-buffers and animated by a simple dynamic simulation with a physically-based approach. As examples they present colored furry doughnuts, cylinders covered with strands, sequences of long hair animation on spheres, and mustaches with different modeling parameters on parallelepipeds.

Anjyo et al. [77] improved a physically based modeling approach to compute the dynamic behavior of hair. For this purpose their hairstyle model consists of three basic steps. The definition of an ellipsoidal hull for the head model. The calculation of hair bending. Finally, the adjustment of the hair style by cutting and modifying hair. This methodology offers satisfactory tools to make different hairstyles based on constraints like gravity and hair collision detection. As examples of design work they present some styles of straight hair. Kurihara et al. [78] propose a method to solve hair animation with collision detection with the body.

Leblanc et al. [79] developed a system to create synthetic hair and fur styles, which consists of two independent modules for modeling and rendering hair. Daldegan et al. [80] describe an integrated system for hair modelling, animation and rendering, based on work from Daldegan et al.[81], LeBlanc et al. [79], Anjyo et al. [77] and Kurihara et al. [78]. Hair rendering is done by raytracing using a modified version of the public domain Rayshade program. An implementation module of the shadow buffer algorithm [82 83] has been added to a raytracing program, based on an earlier version of hair rendering based on pixel blending [79]. The process is step by step. First, the shadow of the scene is calculated for each light source, as well as for the light sources for the hair shadows. The hair shadows are calculated for the object surface and individually for each hair. Finally the hairstyle is blended into the scene, using all shadow buffers. The result is an image with a three-dimensional realistic hairstyle rendering where complex shadow interaction and highlight effects can be seen and appreciated. Color Plate 2 shows an example of a synthetic actress with a hairstyle. Color Plate 3 shows an example of a synthetic actor with hair, beard and skin texture.

7.0 Task-level and Behavioral Animation

7.2 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 [84] 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. Reynolds [85] introduced the term and the concept of behavioral animation in order to describe the automatization of such higher level animation. This new approach of computer animation has less and less to do with the techniques of traditional animation but more and more with the techniques of actor direction. The animator is responsible for the design of the behavior of characters from path planning to complex emotional interactions between characters. 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. The computer director directs at the video screen synthetic actors, decors, lights and cameras using a natural language. If it is in real time, it will be like directing a real film but in a synthetic world. We will enter into the era of real computer-generated films, produced in a virtual world and directed by real human directors.

There are a lot of methods for controlling motion of synthetic actors. For example, Zeltzer [86] classifies animation systems as being either guiding, animator-level or task-level systems. Magnenat Thalmann and Thalmann [1] propose a new 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 it privileged in animating the synthetic actor. For example, in a keyframe system for an articulated body, the privileged information to be manipulated is the angle. In a forward dynamics-based system, the privileged information is a set of forces and torques; of course, in solving the dynamic equations, joint angles are also obtained in this system, but we consider these as derived information. In fact, any motion control method will eventually have to deal with geometric information (typically joint angles), but only geometric motion control methods explicitly privilege this information at the level of animation control..

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. For example, rotoscopy or keyframe systems are typical of systems that manipulate angles. Inverse kinematic methods may be also considered as being in this category. The extension of the principle of kinematic constraints to the imposition of trajectories on specific points of the body is also of geometric nature. With the advent of Virtual Reality devices and superworkstations, brute force methods like rotoscopy-like methods tend to come back.

- 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. Typical physical motion control methods for single actors which consider no other aspect of the environment animate articulated figures through forces and torques applied to limbs. 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 aso 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. A typical example is the definition of a command to impose a degree of fatigue on an actor like suggested by Lee et al. [[87]] in their method of Strength Guided Motion.

As there is no general method applicable to complex motions, only a combination of various techniques may result in a realistic motion with a relative efficiency. Integration of different motion generators is vital for the design of complex motion where the characterization of movement can quickly change in terms of functionality, goals and expressivity. This induces a drastic change in the motion control algorithm at multiple levels: behavioral decision making, global criteria optimization and actuation of joint level controllers. By now, there is no global approach which can reconfigure itself with such flexibility.

7.2 Task-level Animation

As stated by Zeltzer [86], a *task-level animation system* must schedule the execution of motor programs to control characters, and the motor program themselves must generate the necessary pose vectors. To do this, a knowledge base of objects and figures in the environment is necessary, containing information about their position, physical attributes, and functionality. With task-level control, the animator can only specify the broad outlines of a particular movement and the animation system fills in the details. According to Lozano-Perez's [[88]] 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 [[89]] or a natural language interface is the most suitable and popular [[90 91]].
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.

7.3 Walking

For many years there has been a great interest in natural gait simulation. According to Zeltzer [92], 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 [17] use inverse kinematics to interactively define gaits for legged animals. Although Girard's model [18] also incorporates some dynamic elements for adding realism, it is not a truly dynamic approach. Also Bruderlin and Calvert [93] 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 [94] describe an efficient forward dynamic simulation algorithm for articulated figures which has a computational complexity linear in the number of joints. To individualize human walking, Boulic et al. [95] 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 (see Fig.7 and Color Plate 4).

FIGURE 7. Walking

7.4 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 [96]. 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. Fig.8 shows an example.

7.5 Impact of the environment

7.5.1 Introduction

Synthetic actors are moving in an *environment* comprising models of physical objects. Their animation is dependent on this environment and the environment may be modified by these actors. Moreover several synthetic actors may interact with each other. Several very complex problems must be solved in order to render three-dimensional animation involving actors in their environment. They may be classified into the following categories:

FIGURE 8. Object grasping and holding (from the film IAD)

- reaching or avoiding obstacles
- contacts and collisions between rigid objects
- contacts and deformations of deformable objects
- group behaviour (this problem will be discussed in Section 7.5).

7.5.2 Obstacle avoidance

Consider, for example, the problem of walking without collision among obstacles. One strategy is based on the Lozano-Perez algorithm [97]. The first step consists of forming a **visibility graph**. Vertices of this graph are composed of the vertices of the obstacles, the start point S and the goal point G. Edges are included if a straight line can be drawn joining the vertices without intersecting any obstacle. The shortest collision-free path from S to G is the shortest path in the graph from S to G. Lozano-Perez and Wesley describe a way of extending this method to moving objects which are not points. Schröder and Zeltzer [98] introduced Lozano-Perez algorithm into their interactive animation package BOLIO. Breen [99] proposes a technique employing cost functions to avoid obstacles. These functions are used to define goal-oriented motions and actions and can be defined so that the variables are the animated parameters of a scene. These parameters are modified in such a way to minimize the cost function.

7.5.3 Contacts and collisions of rigid objects

The reaction of an actor to the environment may also be considered using dynamic simulation in the processing of interactions between bodies. The interaction is first identified and then a response is generated. The most common example of interaction with the environment is the **collision**. Analytical methods for calculating the forces between colliding rigid bodies have been presented. Moore and Wilhelms [100] modelled simultaneous collisions as a slightly staggered series of single collisions and used non-analytical methods to deal with bodies in resting contact. Hahn [101] prevented bodies in resting contact as a series of frequently occuring collisions. Baraff [102] presented an analytical method for finding forces between contacting polyhedral bodies, based on linear programming techniques. The solution algorithm used is heuristic. A method for finding simultaneous impulsive forces between colliding polyhedral bodies is also described. Baraff [103] also proposed a formulation of the contact forces between curved surfaces that are completely unconstrained in their tangential movement. A collision detection algorithm exploiting the geometric coherence between successive time steps of the simulation is explained. Von Herzen et al. [104] developed a collision algorithm for time-dependent parametric surfaces. Hahn [77] describes the simulation of the dynamic interaction among rigid bodies taking into account various physical characteristics such as elasticity, friction, mass and moment of inertia to produce rolling and sliding contacts.

7.5.4 Deformable and flexible objects

Terzopoulos and Fleischer [105] developed *deformable models* capable of perfectly elastic and inelastic behaviour, viscoelasticity, plasticity, and fracture. The models recently developed by Terzopoulos et al. [58] are for example implemented using the Finite Difference Method, and collisions between elastic objects are simulated by creating potential energy around each object, i.e. intersections between deformable bodies are avoided by surrounding the object surfaces with a repulsive collision force. This is a *penalty method*.

7.6 Behavioral animation

Motion of 3D characters is not simply a matter of mechanics: you cannot walk exactly the same way from the same bar to home twice. Mechanics-based motions 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 the same actions make these characters different for any observer. Behaviour and personality of the human beings are also an essential cause of the observable differences. 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 [85] 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 [106] describe behavioural simulation as a means to obtain global motion by simulating simple rules of behaviour between locally related actors. Lethebridge and Ware [107] 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.

For solving the problem of a synthetic actor crossing a room with furniture (table, chairs etc.), the use of an algorithm like the Lozano-Perez algorithm will certainly provide a trajectory avoiding the obstacle. But this trajectory won't be "natural". No human would follow such a path! The decision of where to pass is based on our vision and we require a certain additional room for comfort. We try to keep a "security distance" from any obstacle. This is a typical behavioral problem that cannot be solved by graph theory or mathematical functions. Moreover, walking depends on our knowledge of the location of obstacles and it is only when we see them that we start to include them in our calculations for adapting the velocity. Renault et al. [108] propose a way of giving to the synthetic actor a vision of his environment. The synthetic environment chosen for these trials is a corridor containing several obstacles of various sizes. The synthetic actor may be placed at any location of the corridor and with any look direction; he will move along the corridor avoiding the obstacles. The system is able to avoid collisions with movable objects, what is not possible with well-known robotics algorithms of path-planning. The model is based on the concept of Displacement Local Automata (DLA), which is an algorithm that can deal with a specific environment. Two typical DLAs are called *follow-the-corridor* and *avoid-the-obstacle*. Vision simulation is the heart of this system. This has the advantage of avoiding all the problems of pattern recognition involved in robotic vision. As input, we have a database containing the description of 3D objects: the environment, the camera characterized by its eye and interest point. As output, the view consists of a 2D array of pixels. each pixel contains the distance between the eye and the point of the object for which this is the projection.

Conclusion

The long-term objective of research in virtual humans is the visualization of the simulation of the behavior of human beings in a given environment, interactively decided by the animator. Behavioral techniques make possible the automating of high-level control of actors such as path planning. By changing the parameters ruling this automating, it is possible to give a different personality to each actor. This behavioral approach should be a major step relatively to the conventional motion control techniques. Our main purpose is the development of a general concept of autonomous actors reacting to their environment and taking decisions based on perception systems, memory and reasoning. The animation system with autonomous actors will be based on the three key components: the locomotor system, the perceptual system, the organism system. With such an approach, we should be able to create simulations of situations such as actors moving in a complex environment they may know and recognize, or actors playing ball games based on their visual and touching perception.

Acknowledgments

The author would like to thank Ronan Boulic, Agnes Daldegan, Prem Kalra, Arghyro Paouri, Hans-Martin Werner and Ying Yang from MIRALab, University of Geneva and Computer Graphics Lab, Swiss Federal Institute of Technology in Lausaane for their contribution to the pictures.

References

[1] Magnenat-Thalmann N, Thalmann D (1991) “Complex Models for Animating Synthetic Actors,” IEEE Computer Graphics and Applications, Vol.11, No5, pp.32-44.

[2] Miller G (1991) MacBounce: A Dynamics-Based Modeler for Character Animation, Proc. Computer Animation '91, Springer-Verlag, Tokyo.

[3] Magnenat-Thalmann N, Thalmann D (1987) The Direction of Synthetic Actors in the Film Rendez-vous à Montréal, IEEE Computer Graphics and Applications , Vol. 7, No12

[4] Magnenat-Thalmann N, Minh HT, de Angelis M, Thalmann D (1989) Design, Transformation and Animation of Human Faces, The Visual Computer, Vol.5, No3, pp.32-39

[5] Barr AH (1984) Global and Local Deformations of Solid Primitives, Proc. SIGGRAPH '84, Computer Graphics, Vol. 18, N0.3, pp.21-30.

[6] Sederberg TW, Parry SR (1986) Free-form deformation of solid geometric models. Proc. SIGGRAPH '86, Computer Graphics, Vol.20, No4, pp.151-160

[7] Coquillart S (1990) Extended Free-Form Deformations: A Sculpturing Tool for 3D Geometric Modeling, Proc. SIGGRAPH '90, Computer Graphics, Vol.24, No.4, pp.187-193.

[8] Nahas M, Huitric H, Saintourens M (1987) Animation of a B-spline figure, The Visual Computer, Vol.3, No5,pp.272-276.

[9] Komatsu, K (1988) Human skin model capable of natural shape variation, The Visual Computer, Vol.3, No5, pp.265-271

[10] Wyvill B, Wyvill G (1989) Field functions for iso-surfaces, The visual Computer, Vol.5, No3

[11] Allan JB, Wyvill B, Witten IA (1989) A Methodology for Direct Manipulation of Polygon Meshes, Proc. Computer Graphics International '89, Leeds, pp.451-469.

[12] LeBlanc A, Kalra P, Magnenat-Thalmann N, Thalmann D (1991) Sculpting With the "Ball & Mouse" Metaphor - Proc. Graphics Interface '91, Calgary, Canada

[13] Paouri A, Magnenat Thalmann N, Thalmann D (1991) Creating Realistic Three-Dimensional Human Shape Characters for Computer-Generated Films, Proc. Computer Animation '91, Springer, Tokyo, pp.89-100.

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

[15] Hollerbach JM, Sahar G (1983) Wrist-Partitioned, Inverse Kinematic Accelerations and Manipulator Dynamics, Intern. Journal of Robotics Research, Vol.2, No4, pp.61-76.

[16] 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.

[17] 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

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

[19] Korein J, Badler NI (1982) Techniques for Generating the Goal-directed Motion of Articulated Structures, IEEE Computer Graphics and Applications, Vol.2, No9, pp.71-81

[20] 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

[21] 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.

[22] 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.

[23] 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.

[24] 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

[25] 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.

[26] 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

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

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

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

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

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

[32] Magnenat-Thalmann N, Laperriere R,d Thalmann D (l988) JLD deformations for hand animation and object grasping, Proc. Graphics Interface'88, Edmonton, Canada.

[33] Gourret JP, Magnenat-Thalmann N, 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

[34] Chadwick J, Haumann DR, Parent RE (1989) Layered Construction for Deformable Animated Characters, Proc. SIGGRAPH '89, Computer Graphics, Vol. 23, No3, pp.234-243

[35] Gascuel MP, Verroust A, Puech C (1991) A Modelling System for Complex Deformable Bodies Suited to Animation and Collision Processing, The Journal of Visualization and Computer Animation, Vol. 2, No3, pp.82-91

[36] Chen DT, Zeltzer D (1992) Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method, Proc. SIGGRAPH'92, Computer Graphics, Vol. 26, No 2, pp.89-98

[37] Delingette H, Watanabe Y, Suenaga Y (1993) Simplex-Based Animation, Proc. Computer Animation '93, Springer, Tokyo.

[38] Turner R, Thalmann D (1993) The Elastic Surface Layer Model for Animated Character Construction, Proc. Computer Graphics International '93, Springer, Tokyo

[39] Ekman P and Friesen W (1978) Facial Action Coding System, Consulting Psychologists Press, Palo Alto.

[40] Parke FI (1975) A Model for Human Faces that allows Speech Synchronized Animation, Computers and Graphics, pergamon Press, Vol.1, No1, pp.1-4.

[41] Parke FI (1982) Parameterized Models for Facial Animation, IEEE Computer Graphics and Applications, Vol.2, No 9, pp.61-68

[42] Platt S, Badler N (1981) Animating Facial Expressions, Proc. SIGGRAPH '81, Computer Graphics, Vol.15, No3, pp.245-252.

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

[44] Magnenat-Thalmann N, Primeau E, Thalmann D (1988), Abstract Muscle Action Procedures for Human Face Animation, The Visual Computer, Vol. 3, No. 5, pp. 290-297.

[45] Terzopoulos D, Waters K (1990) Physically Based Facial Modeling, Analysis, and Animation, Visualization and Computer Animation, Vol. 1, No. 2, pp. 73-80.

[46] Parke FI (1991), Control Parameterization for Facial Animation, Proc. Computer Animation '91, Geneva, Switzerland, (Eds Magnenat-Thalmann N and Thalmann D), pp. 3-13.

[47] deGraf B (1989) in State of the Art in Facial Animation, SIGGRAPH '89 Course Notes No. 26, pp. 10-20.

[48] Williams L (1990), Performance Driven Facial Animation, Proc SIGGRAPH '90, pp. 235-242.

[49] Terzopoulos D, Waters K (1991) Techniques for Facial Modeling and Animation, in: Magnenat Thalmann N, Thalmann D (eds) Computer Animation '91, Springer, Tokyo, pp.59-74

[50] Kurihara T, Arai K (1991), A Transformation Method for Modeling and Animation of the Human Face from Photographs, Proc. Computer Animation '91 Geneva, Switzerland, (Eds Magnenat-Thalmann N and Thalmann D), pp. 45-57.

[51] Kalra P, Mangili A, Magnenat-Thalmann N, Thalmann D (1992) Simulation of Facial Muscle Actions Based on Rational Free Form Deformations, Proc. Eurographics '92, Cambridge, UK.

[52] 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.

[53] Hill DR, Pearce A, Wyvill B (1988), Animating Speech: An Automated Approach Using Speech Synthesised by Rules, The Visual Computer, Vol. 3, No. 5, pp. 277-289.

[54] Kalra P, Mangili A, Magnenat-Thalmann N, Thalmann D (1991) SMILE: a Multilayered Facial Animation System, Proc. IFIP Conference on Modelling in Computer Graphics, Springer, Tokyo, Japan, pp.189-198.

[55] Weil Jerry (1986) The Synthesis of Cloth Objects, Proc. SIGGRAPH'86, Computer Graphics, Vol.20, No.4, 1986, pp.49-54.

[56] Hinds BK, McCartney J (1990) Interactive garment design, The Visual Computer, Vol. 6, pp.53-61.

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

[58] Aono M (1990) A Wrinkle Propagation Model for Cloth, Proc. Computer Graphics International '90, Springer, Tokyo, pp.96-115.

[59] Kunii TL, Gotoda H (1990) Modeling and Animation of Garment Wrinkle Formation Processes, Proc. Computer Animation'90, Springer, Tokyo, pp.131-147.

[60] Lafleur B, Magnenat Thalmann N, Thalmann D (1991) Cloth Animation with Self-Collision Detection. In Proc. IFIP Conference on Modeling in Computer Graphics, Springer, Tokyo, pp.179-187.

[61] Carignan M, Yang Y, Magnenat Thalmann N, Thalmann D (1992) Dressing Animated Synthetic Actors with Complex Deformable Clothes, Proc. SIGGRAPH '92, Computer Graphics, Vol. 26, No.2, pp.99-104.

[62] Yang Y, Magnenat-Thalmann N, Thalmann D (1992) 3D Garment Design and Animation — A New Design Tool For The Garment Industry, Computers in Industry, Vol.19, pp.185-191.

[63] Blinn JF, Newell ME (1976), Texture and Reflection in Computer Generated Images, Comm. ACM, Vol. 19, No. 10, pp. 542-547.

[64] Heckbert PS (1986), Survey of Texture Mapping, IEEE CG&A, Nov. 1986, pp. 56-67.

[65] Oka M, Tsutsui K, Ohba A (1987), Real Time Manipulation of Texture-Mapped Surfaces, Proc. SIGGRAPH '87, Computer Graphics, Vol. 21, No. 4, pp. 181-188.

[66] Nahas M, Huitric H, Rioux M and Domey J (1990), Registered 3D Texture Imaging, Proc. Computer Animation '90, Springer, Tokyo, pp. 81-91.

[67] Waters K, Terzopoulos D (1991), Modeling and Animating Faces using Scanned Data, The Journal of Visualization and Computer Animation, Vol 2, pp. 123-128.

[68] Kurihara T, Arai K (1991), A Transformation Method for Modeling and Animation of the Human Face from Photographs, Proc. Computer Animation '91, Springer, Tokyo, pp. 45-57.

[69] Yau JFS and Duffy ND (1988) A Texture Mapping Approach to 3D Facial Image Synthesis, Computer Graphics Forum, 7, pp. 129-134.

[70] Prem Kalra and Nadia Magnenat-Thalmann, Simulation of Facial Skin using Texture Mapping and Coloration, Proc. ICCG'93, North Holland

[71] Csuri C, Hakathorn R, Parent R, Carlson W, Howard M (1979) Towards an interactive high visual complexity animation system," Computer Graphics, Vol.13, No.2, pp. 289-299.

[72] Perlin K, Hoffert (1989) Hypertexture, Computer Graphics, Vol.23, No.3, pp. 253-262.

[73] Kajiya JT, Kay TL (1989) Rendering Fur with Three Dimensional Textures, Computer Graphics Vol. 23, No. 3, pp. 271-280.

[74] Miller GSP (1988a) From Wire-Frame to Furry Animals, Proc. Graphics Interface, pp. 138-146.

[75] Watanabe Y, Suenaga Y (1989) Drawing Human Hair Using Wisp Model, Proc. Computer Graphics International '89, pp. 691-700.

[76] Rosenblum RE, Carlson WE, Tripp III E (1991) Simulating the Structure and Dynamics of Human Hair: Modelling, Rendering and Animation, The Journal of Visualization and Computer Animation,Vol. 2, No. 4 , pp. 141-148.

[77] Anjyo K, Usami Y, Kurihara T (1992) A Simple Method for Extracting the Natural Beauty of Hair, Computer Graphics, Vol. 26, No. 2, pp. 111-120.

[78] Kurihara T, Anjyo K, Thalmann D (1993) Hair Animation with Collision Detection. In: Models and Techniques in Computer Animation, Springer-Verlag, Tokyo, pp.128-138.

[79] LeBlanc A, Turner R, Thalmann D (1991b) Rendering Hair using Pixel Blending and Shadow Buffers. The Journal of Visualization and Computer Animation 2, 3, pp. 92-97

[80] Daldegan A, Magnenat Thalmann N, Kurihara T, Thalmann DAn Integrated System for Modeling, Animating and Rendering Hair , Proc. Eurographics '93

[81] Daldegan A, Magnenat Thalmann N (1993) Creating Virtual Fur and Hair Styles For Synthetic Actors. In: Communicating with Virtual Worlds, Springer-Verlag, Tokyo, pp.358-370.

[82] Williams L (1978) Casting Curved Shadows on Curved Surfaces, Computer Graphics, Vol.12, No3, pp. 270-274.

[83] Reeves WT, Salesin DH and Cook R.L (1987) Rendering Antialiased Shadows with Depth Maps, Computer Graphics 21(4), pp. 283-291.

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

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

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

[87] Lee P, Wei S, Zhao J, Badler NI (1990) Strength Guided Motion, Proc. SIGGRAPH '90, Computer Graphics, Vol. 24, No 4, pp.253-262

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

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

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

[91] 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

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

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

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

[95] 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.

[96] Rijpkema H, Girard M (1991) Computer Animation of Knowledge-based Human Grasping, Proc. Siggraph '91, Computer Graphics, Vol. 25, No3, pp.339-348

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

[98] Schröder P, Zeltzer D (1988) Pathplanning inside Bolio, in: Synthetic Actors: The Impact of Artificial Intelligence and Robotics on Animation, Course Notes SIGGRAPH '88, ACM, pp.194-207.

[99] Breen DE (1989) Choreographing Goal-Oriented Motion Using Cost Functions, in: Magnenat-Thalmann N, Thalmann D, State-of.the-Art in Computer Animation, Springer, Tokyo, pp. 141-152

[100] Moore M, Wilhelms J (1988) Collision Detection and Response for Computer Animation, Proc. SIGGRAPH'88, Computer Graphics, Vol.22, No.4, pp.289-298

[101] Hahn JK (1988) Realistic Animation of Rigid Bodies, Proc. SIGGRAPH '88, Computer Graphics , Vol. 22, No 4 , pp.299-308.

[102] Baraff D (1989) Analytical Methods for Dynamic Simulation of Non-Penetrating Rigid Bodies, Proc. SIGGRAPH '89, Computer Graphics, Vol. 23, No3, pp.223-232.

[103] Baraff D (1990) Curved Surfaces and Coherence for Non-Penetrating Rigid Body Simulation, Proc. SIGGRAPH '90, Computer Graphics, Vol. 24, No4, pp.19-28.

[104] Von Herzen B, Barr AH, Zatz HR (1990) Geometric Collisions for Time-Dependent Parametric Surfaces, Proc. SIGGRAPH'90, Computer Graphics, Vol.24, No.4, pp.39-46

[105] Terzopoulos D, Fleischer K (1988) Deformable Models, The Visual Computer , Vol.4, No6, pp.306-331

[106] 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

[107] 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

[108] Renault O, Magnenat-Thalmann N, Thalmann D (1990) A Vision-Based Approach to Behavioural Animation, Visualization and Computer Animation Journal, John Wiley, Vol,1, No1 (July 1990)

address: Professor Daniel Thalmann
Computer Graphics Lab
Swiss Federal Institute of Technology
CH 1015 Lausanne
Switzerland
tel: +41-21-693-5214
fax: +41-21-693-5328
e-mail: thalmann@di.epfl.ch

Plate 1: Dressed synthetic actress

Plate 2: Hair rendering

Plate 3: Skin texture, hair, beard and mustache

Plate 4: Still walking