Computer Animation: a Key Issue for Time Visualization

Nadia Magnenat Thalmann
MIRALab, University of Geneva
24, rue du Général-Dufour
CH 1211 Geneva 4, Switzerland
E-mail: thalmann@uni2a.unige.ch

Daniel Thalmann
Computer Graphics Lab
Swiss Federal Institute of Technology
CH 1015 Lausanne, Switzerland
E-mail: thalmann@di.epfl.ch

Abstract

This paper discusses the relationship between Computer Animation and Visualization. It shows how Computer Animation methods may help to understand physical laws by adding motion control to data in order to show their evolution over time. It presents the state-of-the-art in Computer Animation, emphasizes new trends like physics-based animation, behavioral animation and VR-based animation. New advances in research at the University of Geneva and the Swiss Federal Institute of Technology are presented.

keywords: animation, visualization, simulation, motion control, physics-based models, behavioral animation,

Computer animation methods may help to understand physical laws by adding motion control to data in order to show their evolution over time. Animation techniques can help increase the scientific knowledge of phenomena and much experimentation should be done in this direction. It is necessary to move from traditional data analysis to the research of more complex structures and to study their meaning in a temporal evolution. When this evolution is not known from the scientist, it is necessary to use traditional computer animation techniques. Visualization has become an important way of validating new models created by scientists. When the model evolves over time, computer simulation is generally used to obtain the evolution of time, and computer animation is a natural way of visualizing the results obtained from the simulation. By using computer animation, scientists may better understand how various phenomena evolve in space and in time.

There is another trend in the relationship between animation, simulation and visualization: research in Computer Animation tends to find physical models to improve the motion. 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. Then, animation languages, scripted systems and director-oriented systems were developed. In the next generation of animation systems, motion control tends to be performed automatically using A.I. and robotics techniques. In particular, motion is planned at a task level and computed using physical laws. This means that research will tend to find physical models to improve the animation. The main purpose is not a validation of the physical models, but to obtain a graphics simulation of the motion as realistic as possible. We deal with physical objects that are characterized by a shape and dynamics properties. Results have been obtained in modeling rigid objects , deformable and flexible objects (Fig.1) or even set of living creatures and there are examples of their behavior under different circumstances (e.g.

forces, specific environment). As described by Hégron et al.,[1] we may distinguish descriptive models used to reproduce an effect without knowledge about its cause and generative models describing the cause which produces the effects.

Fig.1. Flying papers (from the film Flashback, directors N.M.Thalmann and D.Thalmann, 1990)

Motion control in scientific visualization

Scientists experiment with new models and visualization is a way of validating the designed model. When the model evolves over time, animation is a natural way of representing the results obtained from the model. The scientific world is made of natural phenomena, some of which are not visible. Nevertheless their visualization can help understanding how some features evolve in space and in time. Hence, the main objective of animation of a physical phenomenon is scientific insight

Phenomena modeling is performed through computed simulation based on physical theories which are specific to the discipline (e.g. molecular dynamics, quantum chemistry, fluid dynamics,...). Since these phenomena are usually defined not by a geometric shape, but through a set of parameters, visualization requires a further step to derive one or more "physical models" suited for representation. Sometimes a geometric shape is associated with the phenomena (e.g. wind on a wing of a plane), but even in this case we can decide not to use the geometry in our visualization. Geometry is a physical attribute of the phenomenon like the others attributes.

The application of animation to the scientific world is fairly uncommon except for visualization of output coming from fluid dynamic models or molecular dynamics. This is mainly due to the computational costs, the need for specialized hardware and the lack of knowledge of scientists. However, there is a real trend to use these techniques and we expect that in the near future developing environment will boost strongly the use of animation for science. The more people develop "scientific world" simulations, the more the development of new powerful techniques will be accelerated. There are still open problems. The main question is whether animation techniques can help increase the scientific knowledge of phenomena and much experimentation should be done in this direction. Introducing time into scientific visualization changes the meaning of visualization. First, a narrow connection with the data is lost. Secondly, the researcher has to acquire a synthetic view of the phenomenon in order to be able to deal with the animation. It is necessary to move from traditional data analysis to the research of more complex structures and to study their meaning in a temporal evolution.

State-of-the-Art in Computer Animation

What is Computer Animation ?

Most of the phenomena which may be represented on the screen of a workstation are typically time-dependent. The techniques of computer graphics allow the construction of 3D graphical objects using geometric modeling techniques. Moreover, in a 3D space, scenes are viewed using virtual cameras and they may be lit by synthetic light sources. In order to visualize these phenomena at any given time, it is necessary to know the appearance of the scene at this time and then Computer Graphics techniques allow us to build and display the scene according to viewing and lighting parameters. The problems to solve are how to express time dependence in the scene, and how to make it evolve over time. These problems and their various solutions are part of what is usually understood by the term Computer Animation[2]. This term suggests that computers bring something new to the traditional way of animating. Traditional animation is defined as a technique in which the illusion of movement is created by photographing a series of individual drawings on successive frames of film. The definition is essentially correct if we change the definition of the words *photographing*, *drawings*, and *successive frames*. A definition of computer animation could be: a technique in which the illusion of movement is created by displaying on a screen, or recording on a recording device a series of individual states of a dynamic scene. Our definition corresponds to the various techniques for producing films using a computer: computer-aided animation, real-time animation, procedural animation, and automatic motion control.

Traditional animated films are produced as a sequence of images recorded frame-by-frame. Computer-animated films may be produced using the exact same process. What is different in this case is the way the frames were produced. In a traditional film, frames are drawings created by artists. In a computer-animated film, frames are produced by the computer based on the artist's directives. As we shall discuss later on, directives could vary from directly drawing each frame on a graphics tablet to just giving orders to three-dimensional characters. The results will be the same: a series of images produced by the computer. With powerful graphics workstations, reasonably complex scenes may be rendered in a fraction of second. What is important is that it is possible to see the animation directly on the workstation screen, without recording it. Such real-time animation is more and more popular. In the future, we may expect that workstations will be able to produce more complex images in 1/25 of a second. However, the complexity of images is also increasing very fast. This means that real-time animation and frame-by-frame animation will always exist. However the level of complexity dividing the two types of animation will be much higher in the future.

The main goal of computer animation is to synthesize the desired motion effect which is a mixing of natural phenomenon perception and imagination. The designer conceives the object's dynamic behavior with his mental representation of causality. He imagines how it moves, gets out of shape or reacts when he pushes, presses, pulls or twists it. So, the animation system has to provide the user with motion control tools able to translate his wishes from his own language.

Classifications of animation and motion control methods

Classification of animation systems

There are a lot of methods for controlling motion. For example, Zeltzer[3] classifies animation systems as being either guiding, animator-level or task-level systems. In guiding systems, the behaviors of animated objects are explicitly described. Typical guiding systems are BBOP[4], TWIXT[5] and MUTAN[6]. In animator level systems, the behaviors of animated objects are algorithmically specified. Typical systems are: GRAMPS[7], ASAS[8] and MIRA[9]. Abstraction mechanisms are generally available[10]. In task level systems, the behaviors of animated objects is specified in terms of events and relationships. There is no general-purpose task-level system available now, but it should be mentioned that the TEMPUS[11] are steps towards task-level animation. We should also mention Extensible director-oriented systems like MIRANIM[12].

State variables and evolution laws

To improve computer animation, attention needs to be devoted to the design of evolution laws[13]. Animators must be able to apply any evolution law to the state variables which drive animation. The meaning of the state variables has to be chosen as close as possible to the animator language and has to produce the expected effect. For instance, for curve creation, some interpolating splines allow the animator to modulate certain curve characteristics as global or local tension, continuity and bias which represent a "natural" way to specify curve shapes. For generative models which are materialized models of causality, two problems occur: how to choose model parameter values and how to predict effects a priori. Let us take the example of multibody system animation by dynamics. To control the motion of the system, the animator has to adjust two kinds of parameters: the value of driving forces and torques, and the values of energetic bindings such as rigidity and damping coefficients of springs and absorbers which determine the system reaction to internal and/or external forces. Even if the user knows the parameter magnitudes, with forward dynamics the kinematics behavior of the system is obtained a posteriori, by experience. The animator has to adjust the parameter values step by step after each new set of frames until he gets the desired motion. The best way to use forward dynamics is to implement a real time simulator like a flight or driving simulator. The same problem occurs for object modeling with fractals for which the object shape is known after computation.

Motion control of articulated bodies

An important class of object to be represented in virtual worlds is the articulated body like a virtual animal, a virtual human, a synthetic robot.. Magnenat Thalmann and Thalmann[14] propose a new classification of computer animation scenes involving articulated bodies both according to the method of controlling motion and according to the kinds of interactions these bodies have. The nature of privileged information for the motion control of these articulated bodies 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, we may consider rotoscopy, a method which uses sensors to provide coordinates of specific points of joint angles of a real human for each frame. Also keyframe systems are typical of systems that manipulate angles; from key angles at selected times, they calculate angles for intermediate frames by interpolation. Inverse kinematics methods may be also considered as being in this category. They determine values of joint angles from values of end effectors. The extension of the principle of kinematics 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 physics. 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. Kinematics-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 the modeling of objects as they 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 relating the forces, torques, constraints and the mass distribution of objects. 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. As trajectories and velocities are obtained by solving equations, we may consider **motions as globally controlled**. Functional methods based on biomechanics are also part of this class.
- The third type of animation is called behavioral animation and takes into account the relationship between each object and the other objects. Moreover the control of animation may be performed at a task- level, but we may also consider **the actor as an autonomous creature**. In fact, we will consider as a behavioral motion control method any method consisting in driving the behavior of this actor by providing high-level directives indicating a specific behavior without any other stimulus..

Geometric and Kinematics Computer Animation Methods

In this group of methods, the privileged information is of a geometric or kinematics nature. Typically, motion is defined in terms of coordinates, angles and other shape characteristics or it may specified using velocities and accelerations, but no force is involved. We distinguish the traditional key frame technique in which the animator explicitly specifies the kinematics by supplying keyframes values whose "in-between" frames are interpolated by the computer, and the procedural methods in which the kinematics determination is based on implicit instructions, for instance "inverse kinematics" where the motion of interior links of a chain is computed from the end link trajectory. Descriptive models give almost complete control to the animator, but lack realism when the number of parameters to control is very high (for instance a human body) and when the dynamic behavior is difficult to plan (for instance the bipendulum motion). Although these methods have been mainly concerned with determining the displacement of objects, they may also be applied in calculating deformations of objects. Many geometric deformation techniques from the area of solids modeling are available, such as global deformation[15] and free-form deformations[16]. A variety of surface modeling methods have been proposed for representing deformable animated characters from standard polygonal surface meshes to implicit surfaces such as soft objects[17] and parametric surfaces such as hierarchical B-splines[18]. These geometric techniques provide ease of control and rapid computation but they have little relation to the physical reality of a flesh and blood creature, and therefore tend to lack realism. In particular, they tend to represent characters either as geometric surfaces, or as uniform solids, both of which ignore the complex internal structure of human or animal anatomy.

Physics-based Animation

Dynamic Simulations

Kinematics-based systems are generally intuitive and lack dynamic integrity. The animation does not seem to respond to basic physical facts like gravity or inertia. Only modeling of objects that move under the influence of **forces** and **torques** can be realistic. Forces and torques cause linear and angular accelerations. The motion is obtained by the **dynamic equations of motion**. These equations are established using the forces, the torques, the constraints and the mass properties of objects. A typical example is the motion of an articulated figure which is governed by forces and torques applied to limbs. Methods based on parameter adjustment are the most popular approach to dynamics-based animation and correspond to **non-constraint methods**. There is an alternative: the **constraint-based methods**: the animator states in terms of constraints the properties the model is supposed to have, without needing to adjust parameters to give it those properties. 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.

Non-Constraint-Based Methods

Non-constraint methods have been mainly used for the animation of articulated figures[19] [20]. There are a number of equivalent formulations which use various motion equations: Newton–Euler formulation, Lagrange formulation, Gibbs–Appell formulation, 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 [21].

Constraint-based Methods

Isaacs and Cohen[22] discuss a method of constraint simulation based on a matrix formulation. Joints are configured as **kinematics 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.[23] using **energy constraints**. Using **dynamic constraints**, Barzel and Barr[24] 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[25] extend dynamic constraints to flexible models using reaction constraints and optimization constraints. Witkin and Kass[26] 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. Cohen[27] takes this concept further and uses spacetime window to control the animation interactively. The subdivision of spacetime into a discrete pieces, or Spacetime Window, over which subproblems can be formulated and solved. The sensitivity of highly non-linear constrained optimization to starting values and solution algorithms can thus be controlled to a great extent by the user.

Collision detection and response

In computer animation, collision detection and response are obviously more important. Some works have addressed collision detection and response. Boyse[28] presented algorithms that carry out interference checking among solids and surfaces. Hahn[29] prevented bodies in resting contact as a series of frequently occurring collisions. Baraff[30] 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 described. Baraff[31] 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. [32] developed a collision algorithm for time-dependent parametric surfaces. Moore and Wilhems[33] presented two collision detection algorithms and the response algorithms for the articulated bodies that conserve linear and angular momentum. Shinya and Forgue[34] proposed an interference detection algorithm through rasterization. Terzopoulos et al.[35] and Platt and Barr[25] proposed to surround the surfaces of deformable models by a self-repulsive collision force, this is a *penalty* method. Lafleur et al.[36] 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. Carignan et al.[37] described a new elastic collision response algorithm for cloth animation.

Behavioral Animation

Sensor-based behavioral animation

Behavior is often defined as the way that animals and human beings act and is also often reduced to the meaning of reacting to the environment. This is not sufficient for computer animation and the definition of the Encyclopedia Universalis is more convenient for us: behavior is the frame of continuous flow between a living creature and its environment. Behavior is not only reacting to the environment but should also include the flow of information by which the environment acts on the living creature as well as the ways the creature codes and uses this information. Several authors have described experiments in "behavioral animation." Reynolds[38] introduces a distributed behavioral model to simulate flocks of birds, herds of land animals, and schools of fish. Haumann and Parent[39] describe behavioral simulation as a means to obtain global motion by simulating simple rules of behavior between locally related actors. They developed a test-bed used to create a library of physically behaving actors which can realistically reproduce the motion of flexible objects. Lethebridge and Ware[40] propose a simple heuristically-based method for expressive stimulus-response animation. They model stimulus-response relationships using "behavior functions," which are created from simple mathematical primitives in a largely heuristic manner. Wilhelms[41] proposes a system based on a network of sensors and effectors. Ridsdale[42] proposes a method that guides lower-level motor skills from a connectionist model of skill memory, implemented as collections of trained neural networks.

Production-system-based behavioral animation

Production systems and L-grammars, as introduced by Lindenmayer are very powerful tools for creating images. They can really take advantage of the power of the computer to represent objects in 3D with any perspective. The creative work of the artist consists, in principle, of defining an axiom and production rules. From these initial data, the computer creates the corresponding images with a complexity only dependent on the number of times the productions are applied. The resulting images are generally aesthetically appealing, because they contain a form of order and regularity present in the nature. The theory of these grammars is strongly related to genetics and the growth of living beings. The theory of L-structures has been mainly used for the visualization of the development and growth of living organisms like plants[43], trees[44] and cells.

VR-based animation

With the existence of graphics workstations able to display complex scenes containing several thousands of polygons at interactive speed, and with the advent of such new interactive devices as the SpaceBall, EyePhone, and DataGlove, it is possible to create applications based on a full 3-D interaction metaphor in which the specifications of deformations or motion are given in real-time. This new concepts drastically change the way of designing animation sequences.

We call **VR-based animation techniques** all techniques based on this new way of specifying animation. We also call **VR devices** all interactive devices allowing to communicate with virtual worlds[45 46 47 48] They include classic devices like head-mounted display systems, DataGloves as well as all 3D mice or SpaceBalls. We also consider as VR devices MIDI keyboards, force-feedback devices and multimedia capabilities like real-time video input devices and even audio input devices.

During the creating process, the animator should enter a lot of data into the computer. Table 1 shows VR devices and the corresponding data and applications.

VR-device	**input data**	**application**
DataGlove	positions, orientations, trajectories, gestures, commands,	hand animation
DataSuit	Body positions, gestures	body animation
6D mouse	positions, orientations	shape creation, keyframe
SpaceBall	positions, orientations, forces	camera motion,
MIDI keyboard	multi-dimensional data	facial animation
Stereo display	3D perception	camera motion, positioning
Head-mounted display (EyePhone)	camera positions and trajectories	camera motion
Force transducers	forces, torques	physics-based animation
Real-time video input	shapes	facial animation
Real-time audio input	sounds, speech	facial animation (speech)

Table 1 Applications of VR-devices in Computer Animation

Thalmann[46] proposes a classification of VR-based Methods for Animation:

- **Real-time rotoscopy method.** This is a **method** a method consisting of recording input data from a VR device in real-time allowing to apply at the same time the same data to a graphics object on the screen. For example, when the animator opens the fingers 3 centimeters, the hand on the screen do exactly the same.
- **Real-time direct metaphors.** This is a method consisting of recording input data from a VR device in real-time allowing to produce effects of different nature but corresponding to the input data. There is no analysis of the meaning of the input data. For example, when the animator presses the fourteenth key on a MIDI synthesizer, the synthetic actor's face on the screen opens his mouth depending on the pressure on the key. The relationship between the VR device and the animated motion is not as straightforward as one might think. Usually, some sort of mathematical function or "filter" has to be placed between the raw 3-D input device data and the resulting motion parameters.
- **Real-time recognition-based metaphors.** This is a method consisting of recording input data from a VR device in real-time. The input data are analyzed. Based on the meaning of the input data, a corresponding directive is executed. For example, when the animator opens the fingers 3 centimeters, the synthetic actor's face on the screen opens his mouth 3 centimeters. The system has recognized the gesture and interpreted the meaning.

Advances in Computer Animation at University of Geneva and Swiss Federal Institute of Technology

MIRALab, University of Geneva and the Computer Graphics Lab at the Swiss Federal Institute of Technology in Lausanne have made new advances in research in Computer Animation, especially in Human Animation In this section, we describe several of the recent results classified in geometric animation, physics-basic animation, behavioral animation and VR-based animation.

Geometric and kinematics Computer Animation methods

Although many techniques have been developed for the motion control of objects; they are generally applied to much simpler systems than humans. 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.

For a general **locomotor system**, only a combination of various techniques may result in a realistic motion with a relative efficiency. Consequently, a locomotor system should be based on several integrated methods. Production of a natural looking motion based on the integration of all parts of body using different types of motion control methods is certainly a challenge. In our case, a blending module is associated with the coach-trainee correction method that created in the context of walking. This method allows the kinematics correction of joint-space based motion with respect to Cartesian constraints[49]. In such a way, it is still possible to modify the key frame sequence, a low-level description of motion, for a higher level goal-oriented requirement. This approach will greatly extend the scope of predefined motions (rotoscopy, specialized model, key-framed etc..). A new methodology emerges for motion conception and editing which is centered on the coach-trainee correction method. An interactive tool has been designed for the visualization, editing and manipulation of multiple track sequences. A sequence is associated with an articulated figure and can integrate different motion generators such as walking, inverse kinematics, and keyframing within a unified framework. The TRACK system[50] provides a large set of tools for track space manipulations and Cartesian space corrections. This approach allows an incremental refinement design combining information and constraints from both the track space (usually joints) and the Cartesian space.

Computer modeling and animation of synthetic faces has attained a considerable attention recently. Because the human face plays the most important role for identification and communication, realistic construction and animation of the face are of immense interest in the research of human animation. The ultimate goal of this research would be to model exactly the human facial anatomy and movements to satisfy both structural and functional aspects. 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. For facial deformations, Kalra et al.[51] have extended the concept of Free Form Deformations (FFD) introduced by Sederberg and Parry[16], a technique for deforming solid geometric models in a free-form manner. To improve the "Barbie-like" aspect of virtual humans, Kalra and Magnenat Thalmann[52] 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. Fig.2. shows an example of facial animation with texture mapping.

Fig.2 Facial animation

Physics-based animation

In the area of physics-based models, we describe animation based on dynamics (Fig.3), simulation of the skin and object deformations using finite element theory (Fig.4), cloth animation, hair animation and real-time 3D character animation using an elastic layered model.

Fig.3 Dynamics-based motion

Fig.4 Object deformed by hand using finite element theory

Cloth animation

Work in cloth animation[37,53,54], at the University of Geneva, is based on the fundamental equation of motion as described by Terzopoulos[35] with the damping term replaced by a more accurate one proposed by Platt and Barr[25]. When a collision is detected, we pass through the second step where we act on the vertices to actually avoid the collision. For this collision response, we have proposed the use of the law of conservation of momentum for perfectly inelastic bodies. This means that kinetic energy is dissipated, avoiding the bouncing effect. A dynamic inverse procedure is used to simulate a perfectly inelastic collision. Such collisions between two particles are characterized by the fact that their speed after they collide equals the speed of their centers of mass before they collide.

The constraints that join different panels together and attach them to other objects are very important in our case. Two kinds of dynamic constraints[24] 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.5 shows an example.

Fig.5 Cloth animation

Hair animation

To generate natural hair animation[55], physical simulation must be applied. However, precise simulation including collision response is impractical because of the large number of individual hairs. Therefore, simple differential equations of one-dimensional angular momenta are used as described by Anjyo et al.[56], and a simplified collision detection method using cylindrical representation[57].

Creating and animating a 3D characters using an elastic surface layer model,

A model is described for creating three-dimensional animated characters[58]. In this new type of layered construction technique, called the elastic surface layer model, a simulated elastically deformable skin surface is wrapped around a traditional kinematics articulated figure. Unlike previous layered models, the skin is free to slide along the underlying surface layers constrained by reaction forces which push the surface out and spring forces which pull the surface in to the underlying layers. By tuning the parameters of the physically-based model, a variety of surface shapes and behaviors can be obtained such as more realistic-looking skin deformation at the joints, skin sliding over muscles, and dynamic effects such as squash-and-stretch and follow-through. Since the elastic model derives all of its input forces from the underlying articulated figure, the animator may specify all of the physical properties of the character once, during the initial character design process, after which a complete animation sequence can be created using a traditional skeleton animation technique. A reasonably complex character at low surface resolution can be simulated at interactive speeds so than an animator can both design the character and animate it in a completely interactive, direct-manipulation environment. Once a motion sequence has been specified, the entire simulation can be recalculated at a higher surface resolution for better visual results. Fig.6 shows an example.

Fig.6 Interactive physics-based 3D character with elastic layered surface model

Behavioral animation

Vision-based animation

For solving the problem of a synthetic actor crossing a room with furniture (table, chairs etc.), Renault et al.[59] 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 robotics 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.

L-system-based behavioral animation

Another approach for behavioral animation is based on timed and parameterized L-system with conditional and pseudo stochastic productions[60]. With this software package a user may create any realistic or abstract shape, play with fascinating tree structures and generate any concept of growth and life development in the resulting animation. To extend the possibilities for more realism in the pictures, external forces have been added, which interact with the L-structures and allow a certain physical modeling. External forces can also have an important impact in the evolution of objects. Tree structures can be elastically deformed and animated by time and place dependent vector force fields. The elasticity of each articulation can be set individually by productions. So, the bending of branches can be made dependent of the branches' thickness, making animation more realistic. The force fields too, can be set and modified with productions. Force can directly affect L-structures. It is possible to simulate the displacement of objects in any vector force field dependent on time and position. An object movement is determined by a class of differential equations, which can be set and modified by productions. The mass of the turtle, who represents the object, can be set as well, by using a special symbol of the grammar. This vector force field approach is particularly convenient to simulate the motion of objects in fluids (air, water) as described by Wejchert and Haumann[61].

VR-based animation

Balaguer et al.[62] describe the *Virtuality Builder II* (*VB2*) an object-oriented framework designed to allow rapid construction of applications using a variety of 3D devices and interaction techniques. *VB2* applications are composed of a group of processes communicating through inter-process communication (IPC). A central process manages the model of the virtual world, and simulates its evolution in response to events in the form of IPC messages coming from the processes that encapsulate asynchronous input devices. Sensory feedback to the user can be provided by several output devices. Visual feedback is provided by real-time rendering on graphics workstations, while audio feedback is provided by *MIDI*output and playback of prerecorded sounds. In VB2-based applications, users interact with dynamic models through direct manipulation, gestures, and virtual tools (Fig.7).

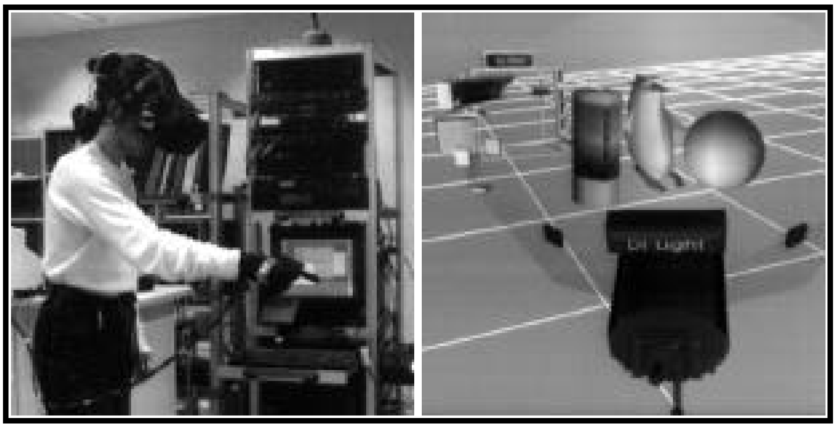

Fig.7. Immersion in Virtual Worlds using Virtuality Builder II

Computer Animation and Time Visualization: new trends for the next years

Computer Animation should not be just considered as a tool to enhance spatial perception by moving the virtual camera or just rotating objects[63]. More sophisticated animation techniques than keyframe animation must be widely used. Computer animation tends to be more and more based on physics and dynamic simulation methods. In the future, the application of computer animation to the scientific world will become very common in many scientific areas: fluid dynamics, molecular dynamics, thermodynamics, plasma physics, astrophysics etc. In the future, real-time complex animation systems will be developed taking advantage of VR-devices and simulation methods.

We strongly believe that *time visualization* will be a major aspect of visualization systems in the future, like volume visualization today. Several authors have already tried to address the problem of time visualization especially to display the evolution of vector fields[64]. Future visualization systems should provide facilities to incorporate the design and editing of evolution laws. These evolution laws could be applied to a large amount of data like particles in motion to visualize flows[65],[66], to complex structures for climate modeling[67], tunneling process in quantum electronic transport[68], or earthquake seismology[69]. We know that volume visualization encompasses both multidimensional image synthesis and multidimensional image processing. Similarly, time visualization will provide for the analysis and understanding of the evolution of 3D volumetric data, for the synthesis of the evolution of 3D volumetric data using appropriate models, and for the 3D interaction with complex time-dependent phenomena. An integration between simulation methods and VR-based animation will lead to systems allowing the user to interact with complex time-dependent phenomena providing interactive visualization and interactive animation. This development will be only possible by developing new approaches to real-time motion. In particular, developments should involve the following concepts: parallelism, neural nets, distributed animation, simplification of structures[70] and pseudo-physics.

Acknowledgments

Researches described in this paper have been sponsored partly by the "Fonds National Suisse pour la Recherche Scientifique" and ESPRIT Project HUMANOID.

References

1 G.Hégron, P.Palamidese and D.Thalmann, "Motion Control in Animation, Simulation and Visualization", Computer Graphics Forum, Vol. 8, No4, 1989, pp.347-352

2 N.Magnenat-Thalmann and D. Thalmann, "Computer Animation: Theory and Practice", Springer Verlag, Tokyo, second edition, 1990.

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

[4] G.Stern, "Bbop- a System for 3D Keyframe Figure Animation", SIGGRAPH '83 Tutorial Notes on Computer Animation, Vol.7, 1983, pp.240-243

[5] J.E.Gomez, "Twixt: a 3-D Animation System", Proc. Eurographics '84, North Holland, 1984, pp.121-134.

[6] D.Fortin, J.F.Lamy and D.Thalmann, "A Multiple Track Animator System for Motion Synchronization", Proc. ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion, Toronto, 1983, pp.180-186

[7] T.J.O'Donnel and J.O.Arthur, "GRAMPS--A Graphics Language Interpreter for Real-Time, Interactive, Three-Dimensional Picture Editing and Animation", Proc. SIGGRAPH'81, Computer Graphics, Vol.15, No3, 1981.

[8] C.W.Reynolds, "Computer Animation with Scripts and Actors", Proc. SIGGRAPH '82, 1982, pp.289-296

[9] N. Magnenat-Thalmann and D.Thalmann, "The Use of High Level Graphical Types in the MIRA Animation System", IEEE Computer Graphics and Applications, Vol. 3, No 9, 1983, pp. 9-16

[10] D.Thalmann and N.Magnenat-Thalmann, "Actor and Camera Data Types in Computer Animation", Proc. Graphics Interface '83, 1983, pp.203-210

[11] N.I.Badler, J.Korein, J.U.Korein, G.M.Radack, L.Shapiro and C.Brown, TEMPUS: A System for the Design and Simulation of Human Figures in a Task-Oriented Environment, The Visual Computer, , Springer International, Vol.1, No4

[12] N.Magnenat-Thalmann, D.Thalmann and M.Fortin, "MIRANIM: An Extensible Director-Oriented System for the Animation of Realistic Images", IEEE Computer Graphics and Applications, Vol. 5, No 3, 1985, pp. 61-73

[13] N.Magnenat Thalmann N and D.Thalmann, "3D Computer Animation: More an Evolution Problem than a Motion Problem", IEEE Computer Graphics and Applications, Vol.5, No 10, 1985, pp. 47-57.

[14] N.Magnenat Thalmann and D.Thalmann, "Complex Models for Animating Synthetic Actors", IEEE Computer Graphics and Applications, Vol.11, No5, 1991, pp.32-44..

[15] A.H.Barr, "Global and Local Deformations of Solid Primitives", Proc.SIGGRAPH '84, Computer Graphics, Vol. 18, No3, 1984

[16] T.W.Sederberg and S.R.Parry, "Free-Form Deformations of Solid Geometric Models", Proc. SIGGRAPH'86 Computer Graphics, Vol. 20, No4, 1986, pp.151-160

[17] J.Bloomenthal and B.Wyvill, "Interactive Techniques for Implicit Modeling", Proc. SIGGRAPH Symposium on Interactive 3D Graphics, Computer Graphics, Vol. 24, No2, 1990, pp.109-116

[18] D.R.Forsey and R.H.Bartels, "Hierarchical B-Spline Refinement", Proc. SIGGRAPH'88, Computer Graphics, Vol. 22, No4, 1988, pp.205-212

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

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

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

[22] P.M.Isaacs and M.F.Cohen, "Mixed Methods for Complex Kinematic Constraints in Dynamic Figure Animation", The Visual Computer, Vol. 4, No6, 1988, pp.296-305.

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

[24] R.Barzel and A.H.Barr, "A Modeling System Based on Dynamic Constraints", Proc. SIGGRAPH '88, Computer Graphics, Vol.22, No4, 1988, pp.179-188

[25] J.C.Platt and A.H.Barr, "Constraint Method for Flexible Models", Proc. SIGGRAPH '88, Computer Graphics , Vol. 22, No4 , 1988, pp.279-288.

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

[27] M.F.Cohen, "Interactive Spacetime Control for Animation" Proc. Siggraph'92, 1992, pp.293-302.

[28] J.W.Boyse, "Interference Detection among Solid and Surfaces", Comm. ACM, Vol.22, No.1, 1979, pp. 3-9.

[29] J.K.Hahn, "Realistic Animation of Rigid Bodies", Proc. SIGGRAPH '88, Computer Graphics, Vol. 22, No 4 , 1988, pp.299-308.

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

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

[32] B.Von Herzen, A.H.Barr and H.R.Zatz, "Geometric Collisions for Time-Dependent Parametric Surfaces", Proc. SIGGRAPH'90, Computer Graphics, Vol.24, No.4, 1990, pp.39-46

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

[34] M.Shinya and M.C.Forgue, "Interference Detection through Rasterization", The Journal of Visualization and Computer Animation, Volume 2 Number 41, 1991, pp.132-134.

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

[36] B.Lafleur, N.Magnenat Thalmann and D.Thalmann, "Cloth Animation with Self-Collision Detection", Proc. IFIP Conf. on Modeling in Computer Graphics, Tokyo, 1991.

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

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

[39] D.R.Haumann and R.E.Parent, "The Behavioral Test-bed: Obtaining Complex Behavior from Simple Rules", The Visual Computer, Vol.4, No 6, 1988, pp.332-347.

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

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

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

[43] P.Prusinkiewicz and A.Lindenmayer, "The Algorithmic Beauty of Plants", Springer-Verlag, 1990.

[44] M.Aono and T.L.Kunii, "Botanical Tree Image Generation", IEEE Computer Graphics and Applications, Vol.4, No5, 1984, pp.10-34

[45] D.Thalmann, "Using Virtual Reality Techniques in the Animation Process", In: Earnshaw R, Gigante M, Jones H (eds) Virtual Reality Systems, Academic Press, 1993.

[46] F.Balaguer and A.Mangili, "Virtual Environments" in: N.Magnenat-Thalmann and D.Thalmann (Eds) New Trends in Animation and Visualization, John Wiley and Sons, 1991, pp.91-106.

[47] FP Brooks Jr , "Walkthrough - A Dynamic Graphics System for Simulating Virtual Buildings", Proc. 1986 Workshop on Interactive 3-D Graphics ACM, 1986, pp.9-22

[48] S.S.Fisher, M.McGreevy, J.Humphries and W.Robinett, "Virtual Environment Display System", Proceeding 1986 Workshop on Interactive 3-D Graphics, ACM, 1986, pp 77-87

[49] R.Boulic R and D.Thalmann, "Combined Direct and Inverse Kinematic Control for Articulated Figures Motion Editing", *Computer Graphics Forum,* Vol. 2, No.4, 1992, pp.189-202.

[50] R.Boulic, Z.Huang, N.Magnenat Thalmann and D.Thalmann, "A Unified Framework for the Motion Manipulation of Articulated Figures with the TRACK System", Proc 3rd Conf. on CAD/CG, Beijing

[51] P.Kalra, A.Mangili, N.Magnenat Thalmann and D.Thalmann, "Simulation of Facial Muscle Actions Based on Rational Free Form Deformations", *Proc. Eurographics '92,* Cambridge, 1992, pp. 59-69.

[52] P.Kalra and N.Magnenat-Thalmann, "Simulation of Facial Skin using Texture Mapping and Coloration", *Proc. ICCG '93*, Bombay, India, 1992, pp.247-256.

[53] H.M.Werner, N.Magnenat Thalmann, D.Thalmann, "User Interface for Fashion Design", Proc. ICCG93, Bombay, North Holland, 1993

[54] N. Magnenat-Thalmann, "Making 3D Clothes for Synthetic Actors" in: Interacting with Images (J.Vince and LMacDonald, eds), 1993, John Wiley

[55] A.Daldegan, N.Magnenat Thalmann, T.Kurihara and D.Thalmann, "An Integrated System for Modeling, Animating and Rendering Hair", Proc. Eurographics '93, Computer Graphics Forum, 1993.

[56] K.Anjyo, Y.Usami and T.Kurihara, "A Simple Method for Extracting The Natural Beauty of Hair", Computer Graphics, Vol. 26 N0. 2, 1992, pp. 11-120.

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

[58] R.Turner and D.Thalmann, "The Elastic Surface Layer Model for Animated Character Construction", Proc. Computer Graphics International '93, Springer, Tokyo, 1993, pp.399-412.

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

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

[61] J.Wejchert and D.Haumann, "Animation Aerodynamics", Proc. SIGGRAPH '91, Computer Graphics, Vol.25, No4, 1991, pp. 19-22.

[62] E.Gobbetti, J.F.Balaguer, D.Thalmann, "VB2: A Framework for Interaction in Synthetic Worlds", Proc. UIST, Atlanta, 1993.

[63] E.J.Farrell, W.C.Yang and R.A.Zappulla, "Animated 3D CT Imaging", IEEE Computer Graphics and Applications, Vol.5, No12, 1985, pp.26-32.

[64] J. Helman and L. Hesselink, "Representation and Display of Vector Field Topology in Fluid Flow Data Sets, Computer, Vol.22, No9, 1989, pp.27-36.

[65] J.Van Wijk, "Flow Visualization with Surface Particles", IEEE Computer Graphics and Applications, Vol.13, N04, 1993, pp.18-24.

[66] A.van Gelder and J. Wilhelms, "Interactive Animated Visualization of Flow Fields", Proc. 1992 Workshop on Volume visualization, ACM Boston, 1992, pp.47-52.

[67] N. Max, R. Crawfis and J. Williams, "Visualization for Climate Modeling", IEEE Computer Graphics and Applications, Vol.13, No4, 1993, pp.34-39.

[68] N.C.Kluksdahl, A.M.Kriman and D.K.Ferry, "The Role of Visualization in the Simulation of Quantum Electronic transport in Semiconductors", Computer, Vol.22, No9, 1989, pp.27-36.

[69] V.Pereyra and J.A.Rial, "Visualizing Wave Phenomenae with Seismic Rays in: G.M.Nielson, B.Shriver and L.Rosenblum, "Visualization in Scientific Computing", IEEE Computer Society Press, Los Alamitos, 1990, pp.174-189.

[70] L.Moccozet and N.Magnenat Thalmann, "Controlling the Complexity of Objects based on Polygonal Meshes", Proc. Computer Graphics International '92, Springer, Tokyo, 1991, pp. 763-780.