Computer Animation in Future Technologies

Nadia Magnenat Thalmann
MIRALab, University of Geneva
Daniel Thalmann
Computer Graphics Lab, Swiss Federal Institute of Technology

Abstract

In this introductory chapter, we try to situate the role of Computer Animation in the new technologies: digital television, virtual reality, multimedia, cooperative work. We also overview the main techniques which will be further discussed throughout this book by international experts.

1 The high-tech role of computer animation

The term "Computer Animation" 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. Is this definition, due to John Halas (Halas and Manwell 1968), still true for Computer Animation ? 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. We use the term "recording" also for photographing and we consider both a cine-camera and a videorecorder as a recording device.

There are two ways of considering computer animation and its evolution. The first approach corresponds to an extension of traditional animation methods by the use of the computer. The second approach corresponds to simulation methods based on laws of physics, especially laws of mechanics. For example, traditional methods allow us to create three-dimensional characters with exaggerated movements while simulation methods are used to try to model a human behavior accurately. For example, consider the bouncing ball example as described by Lasseter (1987). The motion is improved by introducing squash and stretch. When an object is squashed flat and stretches out drastically, it gives the sense that the object is made out of a soft, pliable material. This is a well known trick used by many traditional animators. It does not produce a realistic simulation, but it gives an impression to the viewer. A bouncing ball motion may be also completely simulated by computer using laws of mechanics such as Newton's laws and quantum conservation. Deformations may be calculated by using complex methods like finite element theory (Gourret et al. 1989). No approach is better than the other; it is like comparing a painting and a photograph. Both are representations of a particular world. If we consider character animation, it is easier to create emotions using a keyframe approach than using mechanical laws. Emotional aspects could very well be simulated using a more formal approach, they would require emotion models be incorporated in a physics-based animation system. This means that there is a third way of producing animation using social and behavioral laws on the top of physics-based laws.

Computer animation has been considered for many years as a new media for advertisement and special effects in films. More recently, the fast development of powerful superworkstations has led to new areas like multimedia, interactive games and Virtual Reality. For these new areas, interactive and real-time animation has become a key issue. Traditional television means that the viewer may only decide which program he/she wants to watch. With the new developments of digital and interactive television and multimedia products, the viewer will be more and more able to interact with programs and this will lead to individual programs for each viewer. Real-time animation and autonomous virtual actors are important in the multimedia industry where an interactively use of the functionality means an immediate asset. Each Film- and TV producer will be interested to develop new features and programmes where the public will be involved interactively. The authors, editors and publishers of interactive TV programs, CD-I's and CD-ROM's exploiting increasingly interactivity need real-time animation capabilities.

Historically, we can observe the following evolution: computer animation has started with very simple methods coming from traditional animation and geometry like keyframes. Then, inverse kinematics and dynamics have been imported from robotics leading to complex simulation techniques. At the same time, time-consuming methods for rendering have been also created. Since a few years, computer animation tends to be more and more based on physics and dynamic simulation methods, especially in the areas of deformations and collision detection and response. With the advent of VR-devices and superworkstations, brute force methods like rotoscopy-like methods tend to come back. In the future, real-time complex animation systems will be developed taking advantage of VR-devices and simulation methods. A typical situation will be the real actor with motion captured by sensors (e.g. Flock of Birds) and autonomous actors completely driven by the computer using real-time behavioral simulation with complex physics-based interaction with the environment. Moreover, these complex scenes could be shared between long-distance partners all over the world. This is one aspect of research we are now putting together in the Virtual Life Network (VLnet) (Capin et al. 1995).

2 A guide to advanced computer animation techniques

Computer animation means the creation of motion through a computer. There are several ways to do it: performance animation, keyframing, inverse kinematics, dynamics, task-level animation, behavioral animation. Moreover specific techniques like cloth animation, hair animation and facial animation offer a big challenge.

Performance animation consists of recording the motion by a specific device for each frame and using this information to generate the image by computer. For example, a human walking motion may be recorded and then applied to a computer-generated 3D character. This approach will provide a very good motion, because it comes directly from reality. However, it has severe limitations as for any new motion, it is necessary to record the reality again. Roberto Maiocchi (Chapter 2) discusses in details this type of technique.

A popular alternative to performance animation is the well-known techniques of keyframing. A brief introduction to the topics is presented in Section 1.3. More details may be found in (Magnenat Thalmann and Thalmann 1990).

The use of **inverse-kinematics** 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. Inverse kinematics is discussed by Boulic and Mas (Chapter 3) and an alternative called inverse kinetics is proposed.

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. 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. Hégron et al. (Chapter 4) discuss in details the methods of dynamic simulation.

The coordination of animated objects is a complex problem that is somewhat similar to the **choreography** problem in dance. There are even more similarities and some differences between the ways in which a choreographer composes a dance and an animator composes a sequence of human animation. State of the art animation systems provide the tools for the animator to "work-out" the detailed movements on an interactive 3-d workstation in a way very similar to that in which the choreographer develops a dance with live dancers. The animator must specify the position and posture of each figure for each step in time. Some simple interpolation may be possible, but there is only limited support for developing movement sequences for complex articulated figures. Just as with the live dance, a complicated animation of articulated figures can take weeks or months to complete. Calvert and Mah (Chapter 5) explore our understanding of the compositional and choreographic processes and describe how some computer based tools have been developed to support the process.

The **morphing technique** has recently attracted much attention because of its astonishing effects in producing animation sequences dealing with the metamorphosis of an object into another object over time. While three-dimensional object modeling and deformation is a solution to the morphing problem, the complexity of objects often makes this approach impractical. Image morphing is based on warp generation and transition control methods. The most tedious part of image morphing is to establish the correspondence of features between images by an animator. This is very interesting to notice that this problem is very similar to the problem of correspondence in keyframe animation as briefly explained in Section 1.3. Lee and Shin (Chapter 6) gives an excellent overview of morphing techniques; more details may be found in (Wolberg 1990).

In order to solve the major problems in both modeling and animation of natural shapes and movement, there are different ways than the use of sophisticated physical models to simulate real world action and interaction. we already discuss the performance animation approach. Another way is to take classic film characters, or video of current-day personalities, and produce computer models and animations of them by **automatic analysis of the video or film footage**. Pentland et al. (Chapter 7) survey progress toward producing such automatic modeling and animation systems.

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. **Facial animation** of synthetic actors is not an easy task, it corresponds to the task of an impersonator. Not only the actors should be realistic in static images, but their motion should be as natural as possible, when a series of images is displayed under the form of a film. Kalra (Chapter 8) gives a complete introduction to the latest techniques in this area.

In order to generate realistic virtual actors, this is essential to create for them clothes and hair. **Cloth animation** is a very interesting problem as it involves the complete problems of collision detection, collision response and deformations. Magnenat Thalmann (Chapter 9) discusses in details cloth simulation as well as **hair modelling** for the creation of realistic virtual actors.

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. Task-level motor control is a problem under study by roboticists. **Planning**, described by Badler et al. (Chapter 10), is a key issue in task-level animation. Chapter 10 discusses these aspects in details.

For many Virtual Reality Applications and Telecooperative Work, Virtual Worlds should be also inhabited by virtual people. These people could be actors controlled by the VR participant or completely autonomous agents. In this latter case, there is a need for a way of providing autonomy or artificial smartness to these virtual humans. By autonomy we mean that the actor does not require the continual intervention of a viewer. Smart actors should react to their environment and take decisions based on perception systems, memory and reasoning. Badler et al. (Chapter 10) also discusses some of these aspects and Thalmann (Chapter 11) presents the concepts of complex **virtual sensors** like virtual vision, audition and tactile for fully **autonomous virtual actors**.

3 A brief survey of traditional computer animation methods

For many years, most authors (Hanrahan and Sturman 1985; Parke 1982; Magnenat-Thalmann and Thalmann 1985; Steketee and Badler 1985; Zeltzer 1985) distinguished between three types of computer animation methods: image-based key-frame animation, parametric keyframe animation and procedural animation.

3.1 Image-based keyframe animation

Keyframe animation consists of the automatic generation of intermediate frames, called in-betweens, based on a set of key-frames supplied by the animator. In image-based keyframe animation, the in-betweens are obtained by interpolating the keyframe images themselves. This is an old technique, introduced by Burtnyk and Wein (1971). When corresponding images have not the same number of vertices, it is necessary to add extra vertices. A linear interpolation algorithm produces undesirable effects such as lack of smoothness in motion, discontinuities in the speed of motion and distortions in rotations. Alternate methods have been proposed by Baecker (1969), Burtnyk and Wein (1976), Reeves (1981). The method may be extended to three-dimensional objects. The principle is the same when objects are modeled in wire-frame. However, the technique is much more complex when objects are facet-based, because a correspondence between facets and between vertices must be found. Vertices and facets must be added in order to have the same numbers for both objects. A complete algorithm has been introduced by Hong et al. (1988).

3.2 Parametric keyframe animation

Parametric key-frame animation is based on the following principle: an entity (object, camera, light) is characterized by parameters. The animator creates keyframes by specifying the appropriate set of parameter values at given time, parameters are then interpolated and images are finally individually constructed from the interpolated parameters. Linear interpolation causes first-derivative discontinuities, causing discontinuities in speed and consequently jerky animation. The use of high-level interpolation such as cubic interpolation or spline interpolation is generally necessary. A good method is the Kochanek-Bartels spline interpolation (Kochanek and Bartels 1984) because it allows the curve to be controlled at each given point by three parameters: tension, continuity and bias. A time value should be added to each control point to control the motion. The method is valid for interpolation between scalar values like angles and vector values like positions.

To explain the method, consider a list of points $\mathbf{P_i}$ and the parameter t along the spline to be determined. The point $\mathbf{V}$ is obtained from each value of t from only the two nearest given points along the curve (one behind $\mathbf{P_i}$, one in front of $\mathbf{P_{i+1}}$). But, the tangent vectors $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$ at these two points are also necessary. This means that, we have:

$$\mathbf{V} = \mathbf{THC}^T \quad (1)$$

where T is the matrix [t^3 t^2 t 1], H is the Hermit matrix, and C is the matrix [$\mathbf{P_i}$, $\mathbf{P_{i+1}}$, $\mathbf{D_i}$, $\mathbf{D_{i+1}}$].

Kochanek and Bartels start from the cardinal spline:

$$\mathbf{D_i} = 0.5\,(\mathbf{P_{i+1}}-\mathbf{P_{i-1}}) = 0.5\,[(\mathbf{P_{i+1}}-\mathbf{P_i}) + (\mathbf{P_i}-\mathbf{P_{i-1}})] \quad (2)$$

This equation shows that the tangent vector is the average of the source chord $\mathbf{P_i}-\mathbf{P_{i-1}}$ and the destination chord $\mathbf{P_{i+1}}-\mathbf{P_i}$. Similarly, the source derivative (tangent vector) $\mathbf{DS_i}$ and the destination derivative (tangent vector) $\mathbf{DD_i}$ may be considered at any point $\mathbf{P_i}$.

Using these derivatives, Kochanek and Bartels propose the use of three parameters to control the splines—**tension**, **continuity**, and **bias**.

The tension parameter t controls how sharply the curve bends at a point $\mathbf{P_i}$; the parameter c controls the continuity of the spline at a point $\mathbf{P_i}$ and the direction of the path as it passes through a point $\mathbf{P_i}$ is controlled by the bias parameter b.

Equations combining the three parameters may be obtained:

$$\mathbf{DS_i} = 0.5\,[(1-t)(1+c)(1-b)\,(\mathbf{P_{i+1}}-\mathbf{P_i}) + (1-t)(1-c)(1+b)\,(\mathbf{P_i}-\mathbf{P_{i-1}})] \quad (3)$$

$$\mathbf{DD_i} = 0.5\,[(1-t)(1-c)(1-b)\,(\mathbf{P_{i+1}}-\mathbf{P_i}) + (1-t)(1+c)(1+b)\,(\mathbf{P_i}-\mathbf{P_{i-1}})] \quad (4)$$

A spline is then generated using Eq. (1) with $\mathbf{DD_i}$ and $\mathbf{DS_{i+1}}$ instead of $\mathbf{D_i}$ and $\mathbf{D_{i+1}}$.

3.3 Procedural animation

In this kind of animation, motion is algorithmically described by a list of transformations (rotations, translations etc.). Each transformation is defined by parameters (e.g. an angle in a rotation). These parameters may change during the animation according to any physical law. These laws may be defined using an analytical form or using a complex process such as the solution of differential equations. Control of these laws may be given by programming as in ASAS (Reynolds 1982) and MIRA (Magnenat-Thalmann and Thalmann 1983) or using an interactive director-oriented approach as in the MIRANIM (Magnenat-Thalmann et al 1985) system. As an example of algorithmic animation, consider the case of a clock based on the pendulum law:

$$\alpha = A \sin(\omega t+\varphi) \quad (1.5)$$

A typical animation sequence may be produced using a program such as:

```
create CLOCK (...);
for FRAME:=1 to NB_FRAMES
        TIME:=TIME+1/24;
        ANGLE:=A*SIN (OMEGA*TIME+PHI);
        MODIFY (CLOCK, ANGLE);
        draw CLOCK;
        record CLOCK
        erase CLOCK
```

4 References

Baecker R (1969) "Picture-driven Animation", Proc. AFIPS Spring Joint Comp. Conf., Vol.34, pp.273-288

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

Burtnyk N, Wein M (1976) ”Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation”, Comm. ACM, Vol.19, No10, pp.564-569.

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

Halas J, Manwell R (1968) ”The Technique of Film Animation”, Hastings House, New York

Hanrahan P and Sturman D (1985) ”Interactive Animation of Parametric Models”, The Visual Computer, Vol.1, No4, pp.260-266.

Hong T.M., R.Laperrière, D.Thalmann (1988) ”A General Algorithm for 3-D Shape Interpolation in a Facet-Based Representation”, Proc. Graphics Interface'88, Edmonton.

Kochanek D and Bartels R (1984) ”Interpolating Splines with Local Tension, Continuity and Bias Tension”, Proc. SIGGRAPH '84, Computer Graphics, Vol.18, No3, pp.33-41.

Lasseter J (1987) ”Principles of Traditional Animation Applied to 3D Computer Animation”, Proc. SIGGRAPH '87, Computer Graphics, Vol. 21, No4, pp.35-44

Magnenat-Thalmann and Thalmann D (1985) ”Computer Animation: Theory and Practice”, Springer, Tokyo (2nd edition 1990)

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

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

Parke F.I. (1982) ”Parameterized Models for Facial Animation”, IEEE Computer Graphics and Applications, Vol.2, No 9, pp.61-68

Reeves W.T. (1981) ”Inbetweening for Computer Animation Utilizing Moving Point Constraints”, Proc.SIGGRAPH '81, pp.263-269.

Reynolds C.W. (1982) “Computer Animation with Scripts and Actors”, Proc. SIGGRAPH'82, pp.289-296.

Steketee S.N., Badler N.I. (1985) ”Parametric Keyframe Interpolation Incorporating Kinetic Adjustment and Phrasing Control”, Proc. SIGGRAPH '85, pp. 255-262.

Wolberg G. (1990) “Digital Image Warping”, IEEE Computer Society Press.

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