USING VIRTUAL REALITY TECHNIQUES IN THE ANIMATION PROCESS

Daniel Thalmann
Computer Graphics Lab
Swiss Federal Institute of Technology
CH-1015 Lausanne
Switzerland

Abstract

This paper tries to find the various functions involved in an animation system and how virtual reality techniques and multimedia input could play a role. A classification of VR-based methods is proposed: real-time rotoscopy methods, real-time direct metaphors and real-time recognition-based metaphors. Several examples are presented: 3D shape creation, camera motion, body motion control, hand animation, facial animation. The hardware and software architecture of our animation system is also described.

1. Introduction

The traditional main difficulty in the process of 3D animation is the lack of 3D interaction. Visual feedback, in a typical computer graphics application that requires items to be positioned or moved in 3-D space, usually consists of a few orthogonal and perspective projection views of the same object in a multiple window format. This layout may be welcomed in a CAD system where, in particular, an engineer might want to create fairly smooth and regular shapes and then acquire some quantitative information about his design. But in 3-D applications like 3D animation where highly irregular shapes are created and altered in a purely visual and esthetic fashion, like in sculpting or keyframe positioning, this window layout creates a virtually unsolvable puzzle for the brain and makes it very difficult (if not impossible) for the user of such interfaces to fully understand his work and to decide where further alterations should be made. Moreover good feedback of the motion is almost impossible making the evaluation of the motion quality very difficult.

For a long time, we could observe virtual worlds only through the window of the workstation's screen with a very limited interaction possibility. Today, new technologies may immerse us in these computer-generated worlds or at least communicate with them using specific devices. In particular, 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.

In this paper, 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. 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. In the next Section, we present a summary of these various VR devices. More details may be found in (Balaguer and Mangili 1991; Brooks 1986; Fisher et al. 1986).

2. A survey of VR devices

2.1 Position/orientation measurement

There are two main ways of recording positions and orientations: magnetic and ultrasonic. Magnetic tracking devices have been the most successful and the Polhemus 3Space Isotrack, although not perfect, is the most common one. A source generates a low frequency magnetic field detected by a sensor. The second approach is generally based on a tripod consisting of three ultrasonic speakers set in a triangular position that emits ultrasonic sound signals from each of the three transmitters.

2.1.1 DataGlove

Hand measurement devices must sense both the flexing angles of the fingers and the position and orientation of the wrist in real-time. Currently, the most common hand measurement device is the DataGlove™ from VPL Research. The DataGlove consists of a lightweight nylon glove with optical sensors mounted along the fingers. In its basic configuration, the sensors measure the bending angles of the joints of the thumb and the lower and middle knuckles of the others fingers, and the DataGlove can be extended to measure abduction angles between the fingers. Each sensor is a short length of fiberoptic cable, with a light-emitting diode (LED) at one end and a phototransistor at the other end. When the cable is flexed, some of the LED's light is lost, so less light is received by the phototransistor. Attached to the back is a 3Space Isotrack system (see 2.2.3) to measure orientation and position of the gloved hand. This information, along with the ten flex angles for the knuckles is transmitted through a serial communication line to the host computer.

2.1.2 DataSuit

Much less popular than the DataGlove, the allows to measure the positions of the body. A typical example of the use of the datasuit is the film of Fuji TV: the Dream of Mr. M. In this film, a 3D character approximately performs the same motion as the animator.

2.1.3 6D devices: 6D Mouse and SpaceBall

Some people have tried to extend the concept of the mouse to 3-D. Ware and Jessome (1988) describe a 6D mouse, called a bat, based on the Polhemus 3Space Isotrack. Logitech's 2D/6D mouse is based on a ultrasonic position reference array, which is a tripod consisting of three ultrasonic speakers set in a triangular position, emits ultrasonic sound signals from each of the three transmitters. These are used to track the receiver position, orientation and movement.

In order to address this problem, Spatial Systems designed a 6 DOF interactive input device called the SpaceBall. This is essentially a "force" sensitive device that relates the forces and torques applied to the ball mounted on top of the device. These force and torque vectors are sent to the computer in real time where they are interpreted and may be composited into homogeneous transformation matrices that can be applied to objects. Buttons mounted on a small panel facing the user control the sensitivity of the SpaceBall and may be adjusted according to the scale or distance of the object currently being manipulated. Other buttons are used to filter the incoming forces to restrict or stop translations or rotations of the object.

2.2 MIDI keyboard

MIDI keyboards have been first designed for music input, but it provides a more general way of entering multi-dimensional data at the same time. In particular, it is a very good tool for controlling a large number of DOFs in a real-time animation system. A MIDI keyboard controller has 88 keys, any of which can be struck within a fraction of second. Each key transmits velocity of keystroke as well as pressure after the key is pressed.

2.3 The stereo displays and the head-mounted displays

Binocular vision considerably enhances visual depth perception. Stereo displays like the StereoView option on Silicon Graphics workstations may provide high resolution stereo real-time interaction. StereoView consists of two items—specially designed eyewear and an infrar emitter. The shutters alternately open and close every 120th of a second in conjunction with the alternating display of the left and right eye view on the display—presenting each eye vith an effective 60Hz refresh. The infrar emitter transmits the left/right signal from the IRIS workstation to the wireless eyewear so that the shuttering of the LCS is locked to the alternating left/right image display. As a result, each eye sees a unique image and the brain integrates these two views into a stereo picture.

The EyePhone is a head-mounted display system which presents the rich 3-D cues of head-motion parallax and stereopsis. It is designed to take advantage of human binocular vision capabilities and presents the general following characteristics:

- headgear with two small LCD color screens, each optically channeled to one eye, for binocular vision.

- special optics in front of the screens, for wide field of view
- a tracking system (Polhemus 3Space Isotrack) for precise location of the user's head in real time.

2.4 Force transducers and force feedback

Robinett (1991) describes how a force feedback subsystem, the Argonne Remote Manipulator (ARM) has been introduced into the Head-Mounted Display project at the University of North Carolina in Chapel Hill. The ARM provides force-feedback through a handgrip with all 6 degrees-of-freedom in translation and rotation.

Luciani (1990) reports several force feedback gestual transducers including a 16-slice-feedback touch and a two-thimbles, which is a specific morphology to manipulate flat objects. By sliding the fingers in the two rings, objects can be grasped, dragged. or compressed. Moreover, their reaction can be felt, for instance their resistance to deformation or displacement.

Minsky et al. (1990) study the theoretical problem of force-feedback using a computer controlled joy-stick with simulation of the dynamics of a spring-mass system including its mechanical impedance.

The DataGlove THX™ is a pneumatic tactile feedback glove. The DataGlove TSR™ is lined with force sensitive resistors on its inner surfaces. When real objects are grasped, a distinct pattern of forces generated over the FSR's. A stored proportional pressure pattern thus measured can be replayed on the DataGlove THX. The THX contains twenty pressure pads in the same positions as the input glove FSR's, as well as bend sensors. The DataGlove FBX™, announced by VPL Research in Summer 1991, is a force feedback glove. It is fitted with micro-actuators producing force feedback to multiple fingers.

2.5 Real-time video input

Input video is now a standard tool for many workstations. However, it generally takes a long time (several seconds) to get a complete picture, which makes the tool useless for real-time interaction. For real-time interaction and animation purpose, images should be digitized at the traditional video frame rate. One of the possibilities for doing this is the Living Video Digitizer (LVD) from Silicon Graphics. With the LVD, images are digitized at a frequency of 25 Hz (PAL) or 30 Hz (NTSC) and may be analyzed by the animation program.

2.6 Real-time audio input

Audio input may be also considered as a way of interactively controlling animation. However, it generally implies a real-time speech recognition and natural language processing.

3. The Animation Process

Three-dimensional animation scenes usually contain static objects grouped into a decor and animated objects that change over time according to motion laws. Moreover, scenes are viewed using virtual cameras and they may be lit by synthetic light sources. These cameras and lights may evolve over time as though manipulated by cameramen. In order to create all the entities and motions, coordinate and synchronize them, known collectively as choreography, 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. Scenes involving synthetic actors imply more complex problems to manage. Human-like synthetic actors have a very irregular shape hard to built especially for well-known personalities. Once the initial human shape has been created, this shape should change during the animation. This is a very complex problem to ensure the continuity and the realism of the deformed surfaces. The human animation is very complex and should be split into body motion control and facial animation. Basically a synthetic actor is structured as an articulated body defined by a skeleton. Skeleton animation consists in animating joint angles. There are two main ways to do that: parametric keyframe animation and physics-based animation. An ultimate objective therefore is to model human facial anatomy exactly including its movements to satisfy both structural and functional aspects of simulation.

During the creating process, the animator should enter a lot of data into the computer. The input data may be of various nature:

- **geometric:** 3D positions, 3D orientations, trajectories, shapes, deformations
- **kinematics:** velocities, accelerations, gestures
- **dynamics:** forces and torques in physics-based animation
- lights and colors
- sounds
- commands

The following table shows VR-devices with corresponding input data:

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)

4. A Classification of VR-based Methods for Animation

4.1 Real-time rotoscopy methods

Traditional **rotoscopy** in 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 off-line approach will provide a very good motion, because it comes directly from reality. However, it does not bring any new concept to animation methodology, and for any new motion, it is necessary to record the reality again.

We call a **real-time rotoscopy 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.

4.2 Real-time direct metaphors

We call a **real-time direct metaphor** 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.

An example of traditional metaphor is the puppet control. A puppet may be defined as a doll with jointed limbs moved by wires or strings. Similarly glove-puppets are dolls of which the body can be put on the hand like a glove, the arms and head being moved by the fingers of the operator. In both cases, human fingers are used to drive the motion of the puppet. This is a metaphor, as the

A strange situation that we have experimented consists in driving a virtual hand using the DataGlove. The virtual hand moves the strings of a puppet. When we consider the motion of the virtual hand, it is a typical real-time rotoscopy method, but the animation of the puppet from the DataGlove is a typical real-time direct metaphor.

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.

4.3 Real-time recognition-based metaphors

We call a **real-time recognition-based metaphor** 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.

4.4. The Ball and mouse metaphor

In essence, *motion parallax* consists of the human brain's ability to render a three-dimensional mental picture of an object simply from the way it moves in relation to the eye. Rotations offer the best results because key positions located on the surface move in a larger variety of directions. Furthermore, in a perspective projection, depth perception is further accentuated by the speed in which features flow in the field of view points located closer to the eyes move faster than the ones situated in back. In a 3-D application, if motion parallax is to be used effectively, this implies the need for uninterrupted display of object movements and thus the requirement for hardware capable of very high frame rates. To acquire this depth perception and mobility in a 3-D application, we make use of a SpaceBall.

When used in conjunction with a common 2-D mouse such that the SpaceBall is held in one hand and the mouse in the other, full three-dimensional user interaction is achieved. The SpaceBall device is used to move around the object being manipulated in order to examine it from various points of view, while the mouse carries out the picking and transformation work onto a magnifying image in order to see every small detail in real time (e.g. vertex creation, primitive selection, surface deformations, cloth panel position, muscle action). In this way, the user not only sees the object from every angle but he can also apply and correct transformations from every angle interactively. In order to improve our approach using stereo display, we also use "StereoView".

5. 3D shape creation

5.1 The Sculpting Approach

The operations conducted in a traditional sculpture can be performed by computer for computer generated objects using a sculpting software (Leblanc et al. 1991; Paouri et al. 1991) based on the ball and mouse metaphor. With this type of 3-dimensional interaction, the operations performed while sculpting an object closely resemble traditional sculpting. The major operations performed using this software include creation of primitives, selection, local surface deformations and global deformations.

Typically, the sculpting process may be initiated in two ways: by loading and altering an existing shape or by simply starting one from scratch. For example, we will use a sphere as a starting point for the head of a person and use cylinders for limbs. We will then add or remove polygons according to the details needed and apply local deformations to alter the

shape. When starting from scratch points are placed in 3D space and polygonized. However, it may be more tedious and time consuming.

To select parts of the objects, the mouse is used in conjunction with the SpaceBall to quickly mark out the desired primitives in and around the object. This amounts to pressing the mouse button and sweeping the mouse cursor on the screen while moving the object with the SpaceBall. All primitives (vertices, edges and polygons) can be selected. Mass picking may be done by moving the object away from the eye (assuming a perspective projection) and carefully picking may be done by bringing the object closer.

5.2 Local and global deformations.

These tools make it possible to produce local elevations or depressions on the surface and to even out unwanted bumps once the work is nearing completion. Local deformations are applied while the SpaceBall device is used to move the object and examine the progression of the deformation from different angles, mouse movements on the screen are used to produce vertex movements in 3D space from the current viewpoint. The technique is intended to be a metaphor analogous to pinching, lifting and moving of a stretchable fabric material. Pushing the apex vertex inwards renders a believable effect of pressing a mould into clay. These tools also make it possible to produce global deformations on the whole object or some of the selected regions. For example, if the object has to grow in a certain direction, it can be obtained by scaling or shifting the object on the region of interest

Pentland et al. (1990) describe a modeling system ThingWorld based on virtual sculpting by modal forces. In the current system, the user specifies forces by use of slider controls, which vary the amount of pinching, squishing, bending etc., force.

6. 3D paths for camera motion

6.1 Metaphors for camera control

One of the most important effect in computer-generated films is the virtual camera motion. We may consider several real-time direct metaphors for controlling camera motion. We may separate these metaphors into kinematics-based metaphors and dynamics-based metaphors.

Ware and Osborne (1990) describe three kinematics-based metaphors for moving through environments:

- *the eyeball in hand*: this technique involves the use of the Polhemus as a virtual video camera which can be moved about the virtual scene
- *the scene in hand*: the scene is made to move in correspondence with the bat. It is akin to having an invisible mechanical linkage which

converts all hand translations and rotations into translations and rotations of the scene.
- *the flying vehicle control*: the bat is used as a control device for a virtual vehicle. The virtual environment is perceived from this vehicle.

Other kinematics-based metaphors have been tested in our laboratory:

- *the virtual sphere metaphor*: the virtual camera is considered as placed on the surface of a sphere centered on the interest point and with a variable radius. The Polhemus is used to control the sphere rotation and the translation is performed using a dial.
- a *variant of the flying vehicle control* consists of positioning the camera on a plane with the normal controlled by the pen of the Polhemus placed on the animator's head.
- *the airplane metaphor*: the camera is considered as always moving forward, the mouse allows to rotate around horizontal axes, rotations around the vertical axis is performed using a dial. The velocity of the camera is controlled by another dial. The Polhemus is used to control the view direction. This metaphor allows the displacement of the camera in one direction while looking at another direction.

Mackinlay et al. (1990) propose the key idea to have the animator indicate a point of interest (target) on a 3D object and use the distance to this target to move the viewpoint logarithmically by moving the same relative percentage of distance to the target on every animation cycle.

6.2 Kinematics and dynamics direct metaphors for camera control

For non-physically-based motion control, we developed ANIMATOR, a 3D interactive program allowing the creation of animation of several entities: objects, cameras and lights. For each entity, a 3D path may be interactively generated using the SpaceBall. The trajectory is generated using a spline. The animator may built the complete hierarchy of entities using only the mouse, then the animation is created by defining paths. These paths may be generated in 3D using the SpaceBall. Time information is defined by control points. The trajectory is then generated using B-splines.

Turner et al. (1991) describe how naturalistic interaction and realistic-looking motion is achieved by using a physically-based model of the virtual camera's behavior. The approach consists to create an abstract physical model of the camera, using the laws of classical mechanics to simulate the virtual camera motion in real time in response to force data from the various 3-D input devices. The behavior of the model is determined by several physical parameters such as mass, moment of inertia, and various friction coefficients which can all be varied interactively, and by constraints on the camera's degrees of freedom which can be simulated by setting certain friction parameters to very high values. This allows us to explore a continuous range of physically-based metaphors for controlling the camera motion. A physically-based camera control model provides a powerful, general-purpose metaphor for controlling virtual cameras in interactive 3-D environments. When used

with force-calibrated input devices, the camera metaphor can be reproduced exactly on different hardware and software platforms, providing a predictable standard interactive "feel". Obviously, pressure-sensitive input devices are usually more appropriate because they provide a passive form of "force-feedback". In our case, the device that gave the best results is the SpaceBall. Plate 1 shows how to inspect an object.

6.3 A real-time rotoscopy method for camera control

For a car moving across a city, one good approach is the use of a SpaceBall to drive the camera. But, this type of approach is not necessarily appropriate for somebody walking in an apartment. The use of the EyePhone allows the animator to really live the scene in which he enters. By recording the position/orientation of the sensor, we get the trajectory. For example, to pan across the virtual world the user should just turn his head.

6.4 Coupling virtual-real camera

A related research is the dynamic coupling between a virtual camera and a real camera. For example, Fellous (1991) explains how the position/orientation of a real camera may be sent to a workstation and used as the current position/orientation of the virtual camera. There is no direct application in pure computer animation, but this may solve the key problem for mixing real and synthetic images.

7. Skeleton animation

One of the most important categories of figures in Computer Animation is the articulated figure. There are three ways of animating these linked figures:

1. by recreating the tools used by traditional animators
2. by simulating the physical laws which govern motion in the real world
3. by simulating the behavioral laws which govern the interaction between the objects.

The first approach corresponds to methods heavily relied upon by the animator: rotoscopy, parametric keyframe animation. The second way guarantees a realistic motion by using kinematics and dynamics (see Plate 2). The problem with this type of animation is controlling the motion produced. 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.

In summary, at a time t, the methodology of calculating the 3D scene is as follows:

1. for a keyframe system: by interpolating the values of parameters at given key times

2. in a dynamic-based system: by calculating the positions from the motion equations obtained with forces and torques
3. for a behavioral animation system: by automatic planning of the motion of an object based on information about the environment (decor and other objects).

From an 3D input point-of-view, in a key-frame systems positions may be entered using for example a SpaceBall and angles may be calculated by inverse kinematics. In a dynamic-based systems, the natural type of input data are forces and torques. For behavioral animation, natural language is the most natural interface and an audio-input may improve the communication.

8. Hand motion

Hand motion is a specific case of animation of articulated bodies. In this section, we present two very different uses of a DataGlove that we have experimented in our laboratory.

8.1 A real-time rotoscopy method for hand animation

This gesture-oriented animation system (Mato Mira 1991) consists basically of two programs: GESTURE LAB, which enables an animator to record, playback and edit real hand movements using a DataGlove, and VOGE (VOice + GEsture), a program which accepts a monologue script consisting of phrases, emphasis and rhythm parameters, and gesture names to generate an animation sequence in which lip and hand movements are synchronized according to the specification. The output of VOGE is fed into the human animation system to obtain the final animation scene. The DataGlove is used in GESTURE LAB to measure the angles from the metacarpophalangeal joints. The user can select a previously recorded sequence to be played, and insert a part of this sequence in a new one, or perform a live recording using the DataGlove. There is a possibility of setting timers to have precise control over the duration and starting point of a playback or live performance. Even if the current version of GESTURE LAB only allows the recording of the performance of a hand, nothing would disallow its extension for full body movement sampling using a DataSuit for example.

8.2 Hand gesture recognition

The purpose of hand gesture recognition is the association of meanings to the various configurations of the hand and its movements. A current approach in our laboratory is the use of a learning process to obtain this type of recognition. There are two stages in the process of recognition:

- the recognition of static hand configurations;

- the recognition of movements considered as series of configurations over time.

For these recognition processes, an efficient way consists of using Kohonen neural networks, because of their efficiency in recognition tasks, their ability to learn to recognize and the possibility to take advantage of the parallelism. To classify postures (static configurations) MLP (Multi-Layer Perception) neural networks may be applied in order to provide a correspondance between the activations of the neurons of the Kohonen network and the types of gestures associated. However, the most important and difficult aspect is the recognition of gestures. Gestures are temporal sequences of hand configurations. Their recognition is a more complex task than posture recognition because it is necessary to take into account the motion history. Neural networks are also a natural way of solving this problem.

9. Facial animation

9.1 A Facial Animation System

Computer simulation of human facial expressions requires an interactive ability to create arbitrary faces and to provide a controlled simulation of expressions on these faces. A complete simulation system should ensure synchronization of eye motion, expression of emotion and word flow of a sentence, as well as synchronization between several actors. A complete example is our SMILE facial animation system (Kalra et al. 1991). This system is based on a methodology for specifying facial animation based on a multi-layered approach. Each successive layer defines entities from a more abstract point of view, starting with muscles deformations, and working up through phonemes, words, sentences, expressions, and emotions.

9.2 Direct Control of Muscular Deformations

At the lowest level, to simulate the muscle action on the skin surface of human face, we developed a 3-D interactive system (see Plate 3) to define regions on the face mesh which correspond to the anatomical description of the facial region on which a muscle action is desired. Plate 4 shows an example. In this system based on the Ball and Mouse metaphor, a parallelepiped control unit can be defined on the region of interest. The deformations which are obtained by actuating muscles to stretch, squash, expand and compress the inside volume of the facial geometry, are simulated by displacing the control point and by changing the weights of the control points of the control-unit. The region inside the control-unit deforms like a flexible volume, corresponding to the displacement and the weights of the control points. Displacing a control point has analogy of adding a muscle vector to the control-unit. Specifying the displacement of the control point is however, more intuitive and simpler to simulate the muscle vectors. In addition, the result matches to the natural notion of muscles acting on that region. For example, a depressor muscle would need squashing the control point inside the control-unit, and a pulling muscle would be interpreted as pulling the control points away from the control-unit. In order to propagate the deformations of regions to the adjoining regions, linear interpolation can be used to decide the

deformation of the boundary points. Higher order interpolation schemes can also be used, however, the movements in the face are rather small, higher ordered discontinuities arising may not affect so much the visual aspects.

For the facial deformations, we use Rational Free Form Deformations (RFFD) (Kalra et al. 1992), which is a an extension of Free Form Deformation (FFD), a technique for deforming solid geometric models in a free form manner (Sederberg and Parry, 1986).

9.3 Facial Animation Control using a MIDI synthesizer or DataGlove

Direct manipulation of the muscular deformations may be also performed in real-time using multi-dimensional data input devices. The two best devices for this purpose are the MIDI keyboard and the DataGlove.

9.4 Video and speech input

Another approach consists of recording a real human face using a video input like the Living Video Digitizer and extracting from the image the information necessary to generate similar facial expressions on a synthetic face. This is a typical rotoscopy method. The problem with this approach is that the image analysis is not easy to perform in real-time. We are experimenting such an approach using snakes (Terzopoulos and Waters 1991).

To specify sentences to be pronounced by a virtual actor, a direct speech input is certainly a good solution.

10. Physics-based Animation and force-feedback devices

10.1 Dynamic Simulation

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. There are three advantages of introducing dynamics into animation control:

- 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: the motion is too regular and it is time-consuming. But the main problem is a problem of user interface. Typically, a hierarchy is built first, then internal parameters are set until the desired effect is obtained. This is a severe limitation, because it means that systems are hard for the animator to control. Parameters (e.g. forces or torques) are sometimes very difficult to adjust, because they are not intuitive. For a car, it is easy to choose the parameters of a spring or of a shock absorber, it is more difficult to adjust those used to simulate the equivalent forces and torques produced by muscle contractions and tensions in an animated figure. The animator does not think in terms of forces or torques to apply to a limb or the body in order to perform a motion.

10.2 Using Force-Feedback Systems in Dynamic Simulation

A solution to the above problem is to introduce a way of communicating forces and torques to the virtual object. However, force prescribed systems will be effective when forces transducers are intensively introduced in computer animation.

Consider, for example, the problem of modeling the deformations of human flesh due to contact with objects. Gourret et al. (1989) propose a method based on finite element theory for calculating deformations when a hand grasps and presses a ball. The natural way of entering hand forces into the system is by using a force-feedback system.

Iwata (1990) describes a force-feedback system with a 9 degree-of-freedom manipulator developed as a tactile input device with reaction force generator. A solid model handler has been developed for manipulating virtual objects using the manipulator. A virtual hand is displayed corresponding to the manipulator. The contact of the virtual hand with virtual objects is detected at 16 control points. The generated forces are calculated according to the solidity of the captured object.

11. A VR-based Animation System

11.1 The Animation system

We are currently developing in cooperation with MIRALab, University of Geneva a complete three-dimensional animation system based on VR-animation. The main objective of the project is the animation of synthetic actors in their environment. In particular, the following animation applications are being developed:

- animation of articulated bodies based on mechanical laws
- vision-based behavioral animation

- hair rendering (LeBlanc et al. 1991) and animation (see Plate 5)
- object grasping
- facial animation
- personification in walking models (see Plates 6 and 7)
- synchronization in task-level animation
- deformation of flexible and elastic objects
- cloth animation (Carignan et al. 1992) with detection of collision (see Plate 8)

In the current version, several applications provide a user interface based on VR devices:

- the sculpting program SURFMAN
- the Muscle and Expression editor in the SMILE Facial Animation system
- the program to create 3-D paths for cameras, objects and light sources
- the hand gesture recording system in GESTURE LAB
- a communication program animator-actor (in development)

The first two programs, are mainly based on the ball and mouse metaphor described in the next Section. SURFMAN may also take advantage of StereoView and the 3-D Polhemus digitizer. Hand gestures are recorded using the DataGlove and 3-D paths are mainly generated using the SpaceBall. We are developing a way of creating camera paths based on the EyePhone. The communication program animator-actor uses the Living Video Digitizer to capture the animator face and the DataGlove for gesture recognition.

11.2 Hardware and software

Our Lab consists in a network of Silicon Graphics IRIS, a NeXT computer and VR devices including EyePhone, DataGloves, SpaceBalls, StereoView, a Polhemus digitizer, video input/output and a MIDI and audio equipment. As shown in Fig.1, every piece of equipment is now connected to the network. Research is conducted in cooperation with the MIRALab group, at the University of Geneva, who possesses about the same equipment with an additional Living Video Digitizer. The software is based on the 5th Dimension toolkit (Russell et al. 1990) and a InterProcess Communication (IPC) library developed in the laboratory and build on top of the standard BSD Unix socket interprocess communication routines. The 5D toolkit implements a traditional 2D widget set integrated with a set of 3D graphical classes and non-conventional input device classes for encapsulating the behavior of the various input devices. An interactive interface builder has been also developed. The IPC library allows all devices and processes in the laboratory to communicate with each other in real time.

12. Conclusion

Since a few years, computer animation tends to be more and more based on physics and dynamic simulation methods. 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. 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 (Moccozet and Magnenat Thalmann 1991) and pseudo-physics.

Acknowledgments

References

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

Brooks F.P. Jr (1986) Walkthrough - A Dynamic Graphics System for Simulating Virtual Buildings Proceedings 1986 Workshop on Interactive 3-D Graphics ACM, pp.9-22

Carignan M, Yang Y, Magnenat Thalmann N, Thalmann D (1992) Dressing Animated Synthetic Actors with Complex Deformable Clothes, Proc. SIGGRAPH '92.

Fellous A (1991) Synthetic TV, Proc. Imagina '91, INA, pp.5.47-5.52.

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

Gourret JP, Magnenat Thalmann N and Thalmann D (1989) Simulation of Object and Human Skin Deformations in a Grasping Task, Proc. SIGGRAPH '89

Iwata H (1990) Artificial Reality with Force-feeedback: Development of Desktop Virtual Space with Compact Master Manipulator, Proc. SIGGRAPH'90, Computer Graphics, Vol.24, No4, pp.165-170.

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

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.

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

LeBlanc A, Turner R, Magnenat Thalmann N (1991) Rendering Hair using Pixel Blending and Shadow Buffers, Journal of Visualization and Computer Animation, Vol.2, No3.

Luciani A (1990) Physical Models in animation: Towards a Modular and Instrumental Approach, Proc. 2nd Eurographics Workshop on Animation and Simulation, Lausanne, Swiss Federal Institute of Technology, pp.G1-G20.

Mackinlay J.D., Card S.K, Robertson G. (1990) Rapid Controlled Movement Through a Virtual 3-D Workspace, Computer Graphics, Vol.24, No4, pp.171-176

Mato Mira F (1991) ICSC World Laboratory LAND-5 Project, Computer Graphics Lab, Swiss Federal Institute of Technology, Lausanne, Switzerland

Minsky M, Ouh-young M, Steele O, Brooks FP Jr, Behensky M (1990) Feeling and Seeing: Issues in Force Display, Proceedings 1990 Workshop on Interactive 3-D Graphics, ACM Press, pp. 235-243.

Moccozet L, Magnenat Thalmann N (1991) Controlling the Complexity of Objects based on Polygonal Meshes, Proc. Computer Graphics International '92, Springer, Tokyo

Paouri A, Magnenat Thalmann N, Thalmann D (1991) Creating Realistic Three-Dimensional Human Shape Characters for Computer-Generated Films, Proc. Computer Animation '91, Geneva, Springer-Verlag, Tokyo

Pentland A, Essa I, Friedmann M, Horowitz B, Sclaroff S (1990) The ThingWorld Modeling system: Virtual Sculpting by Modal Forces, Proceedings 1990 Workshop on Interactive 3-D Graphics, ACM Press, pp. 143-144.

Robinett W (1991) Head-Mounted Display Project, Proc. Imagina '91, INA, pp.5.5-5.6

Sederberg TW, Parry SR (1986) Free-form Deformation of Solid Geometric Models, Proc.SIGGRAPH'86, Computer Graphics, Vol.20, No4, pp.151-160.

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

Turner R, Gobbetti E, Balaguer F, Mangili A, Thalmann D, Magnenat-Thalmann N (1990) An Object-Oriented Methodology Using Dynamic Variables for Animation and Scientific Visualization, in: Chua TS, Kunii TL, CG International '90, Springer, Tokyo, pp.317-328.

Turner R, Balaguer F, Gobbetti E, Thalmann D (1991), Physically-Based Interactive Camera Motion Control Using 3-D Input Devices, in: Patrikalakis N (ed.) Scientific Visualization of Physical Phenomena, Springer, Tokyo, pp.135-145.

Ware C, Jessome DR (1988) Using the Bat: A Six Dimensional Mouse for Object Placement, Proc. Graphics Interface '88, pp.119-124.

Ware C, Osborne S, "Exploration and Virtual Camera Control in Virtual Three Dimensional Environments", Computer Graphics, 24 (2), pp. 175-183.

Daniel Thalmann is currently full Professor, Head of the Computer Science Department, and Director of the Computer Graphics Laboratory at the Swiss Federal Institute of Technology in Lausanne, Switzerland. He is also Adjunct Professor at the University of Montreal, Canada. Since 1977, he was Professor at the University of Montreal and codirector of the MIRALab research laboratory. He received his diploma in nuclear physics and Ph.D in Computer Science from the University of Geneva. He is coeditor-in-chief of the *Journal of Visualization and Computer Animation*, member of the editorial board of the *Visual Computer* and *CADDM Journal* and cochairs the EUROGRAPHICS Working Group on Computer Simulation and Animation. Daniel Thalmann's research interests include 3D computer animation, virtual reality and image synthesis. He has published more than 100 papers in these areas and is coauthor of several books including: *Computer Animation: Theory and Practice* and *Image Synthesis: Theory and Practice*. He is also codirector of several computer-generated films.

Plate 1: Inspection of an object using a physics-based camera

Plate 2: Dynamics-based motion

Plate 3: The SMILE facial animation system

Plate 4: Deforming muscular regions

Plate 5: Face shape obtained by 3D with hair rendering

Plate 6: Individualized walking model

Plate 7: Walking along an arbitrary path

Plate 8: Fashion show