Virtual Reality Software and Technology

Nadia Magnenat Thalmann
MIRALab, Centre Universitaire d'Informatique
University of Geneva
24, rue du Général-Dufour
CH-1221 Geneva 4, Switzerland
fax: +41-22-320-2927
E-mail: thalmann@cui.unige.ch

Daniel Thalmann
Computer Graphics Lab
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland
fax: +41-21-693-5328
E-mail: thalmann@lig.di.epfl.ch

1 Foundations of Virtual Reality

Virtual Reality (VR) refers to a technology which is capable of shifting a subject into a different environment without physically moving him/her. To this end the inputs into the subject's sensory organs are manipulated in such a way, that the perceived environment is associated with the desired Virtual Environment (VE) and not with the physical one. The manipulation process is controlled by a computer model that is based on the physical description of the VE. Consequently, the technology is able to create almost arbitrarily perceived environments.

Immersion is a key issue in VR systems as it is central to the paradigm where the user becomes part of the simulated world, rather than the simulated world being a feature of the user's own world.

The first “immersive VR systems” have been the flight simulators where the immersion is achieved by a subtle mixture of real hardware and virtual imagery.

The term "immersion" is a description of a technology, which can be achieved to varying degrees. A necessary condition is Ellis' notion [1] of a VE, maintained in at least one sensory modality (typically the visual). For example, a head-mounted display with wide field of view, and at least head tracking would be essential. The degree of immersion is increased by adding additional, and consistent modalities, greater degree of body tracking, richer body representations, decreased lag between body movements and resulting changes in sensory data, and so on.

Astheimer [2] defines **immersion** as the feeling of a VR user, that his VE is real. Analogously to Turing's definition of artificial intelligence: if the user cannot tell, which reality is "real", and which one is "virtual", then the computer generated one is immersive. A high degree of immersion is equivalent to a realistic VE. Several conditions must be met to achieve this: the most important seems to be small feedback lag; second is a wide field-of-view. Displays should also be stereoscopic, which is usually the case with head-mounted displays. A low display resolution seems to be less significant.

According to Slater [3], an Immersive VE (IVE) may lead to a sense of presence for a participant taking part in such an experience. **Presence** is the psychological sense of "being there" in the environment based on the technologically founded immersive base. However, any given immersive system does not necessarily always lead presence for all people. Presence is so fundamental to our everyday existence that it is difficult to define. It does make sense to consider the negation of a sense of presence as the loss of locality, such that "no presence" is equated with no locality, the sense of where self is as being always in flux.

2 VR devices

2.1 Magnetic position/orientation trackers

The main way of recording positions and orientations: is to use magnetic tracking devices as those manufactured by Polhemus and Ascension Technology. Essentially, a source generates a low frequency magnetic field detected by a sensor.

For example, Polhemus STAR*TRAK® is a long range motion capture system that can operate in a wireless mode (totally free of interface cables) or with a thin interconnect cable. The system can operate in any studio space regardless of metal in the environment, directly on the studio floor. ULTRATRAK® PRO is a full body motion capture system, it is also the first turnkey solution developed specifically for performance animation. ULTRATRAK PRO can track a virtually unlimited number of receivers over a large area. FASTRAK, an award-winning system is a highly accurate, low-latency 3D motion tracking and digitizing system. FASTRAK can track up to four receivers at ranges of up to 10 feet. Multiple FASTRAKs can be multiplexed for applications that require more than four receivers.

Ascension Technologies manufactures several different types of trackers including the MotionStar Turn-key, the motionStar Wireless, and the Flock of Birds.

MotionStar Wireless was the first magnetic tracker to shed its cables and set the performer free. Motion data for each performer is now transmitted through the air to a base station for remote processing. We've combined our world famous MotionStar DC magnetic tracker with the best wireless technology to give real-time untethered motion capture. There is absolutely no performance compromise. Twist, flip, and pirouette freely without losing data or getting tied up in knots.

MotionStar® Turn-key is a motion-capture tracker for character animation. It captures the motions of up to 120 receivers simultaneously over long range without metallic distortion. Each receiver is tracked up to 144 times per second to capture and filter fast complex motions with instantaneous feedback. Utilizes a single rack-mounted chassis for each set of 20 receivers.

Flock of Birds® is a modular tracker with six degrees of freedom (6DOF) for simultaneously tracking the position and orientation of one or more receivers (targets) over a specified range of ±4 feet. Motions are tracked to accuracies of 0.5° and 0.07 inch at rates up to 144Hz. The Flock employs pulsed DC magnetic fields to minimize the distorting effects of nearby metals. Due to simultaneous tracking, fast update rates and minimal lag occur even when multiple targets are tracked. Designed for head and hand tracking in VR games, simulations, animations, and visualizations.

DataGloves

Hand measurement devices must sense both the flexing angles of the fingers and the position and orientation of the wrist in real-time. The first commercial hand measurement device was the DataGlove® from VPL Research. The DataGlove® (Figure 1) 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 Polhemus sensor 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.

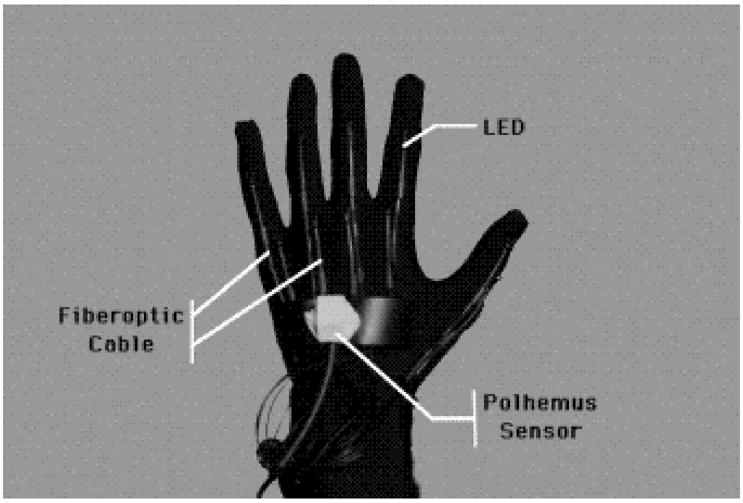


Figure 1. The DataGlove®

CyberGlove® of Virtual Technologies is a lightweight glove with flexible sensors which accurately and repeatably measure the position and movement of the fingers and wrist. The 18-sensor model features two bend sensors on each finger, four abduction sensors, plus sensors measuring thumb crossover, palm arch, wrist flexion and wrist abduction. Many applications require measurement of the position and orientation of the forearm in space. To accomplish this, mounting provisions for Polhemus and Ascension 6 DOF tracking sensors are available for the glove wristband.

3D Mouse and SpaceBall®

Some people have tried to extend the concept of the mouse to 3-D. Ware and Jessome [4] describe a 6D mouse, called a bat, based on a Polhemus tracker.

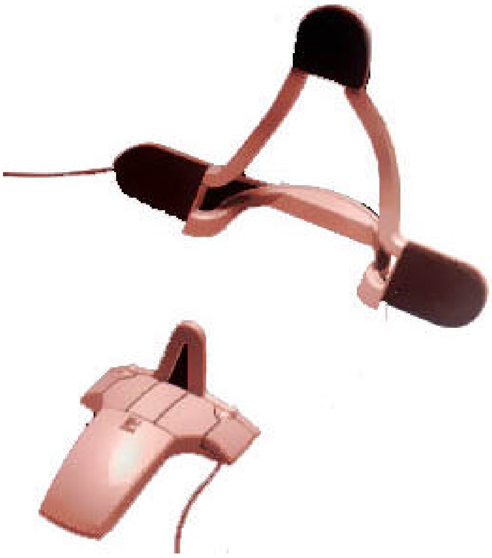


Figure 2. Logitech 3D mouse

The Logitech 3D mouse (Figure 2) 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. It provides proportional output in all 6 degrees of freedom: X, Y, Z, Pitch, Yaw, and Roll.

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. Figure 3 shows a SpaceBall®

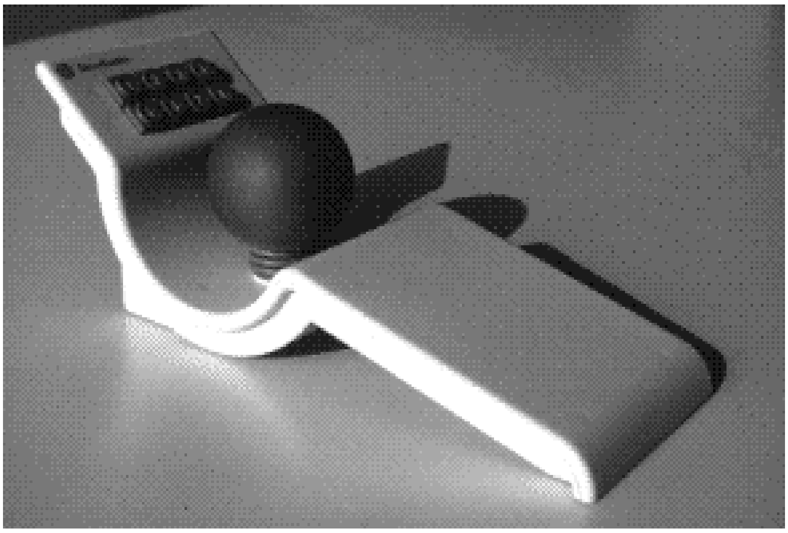


Figure 3. SpaceBall®.

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.

Shutter glasses

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 infrared 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 with an effective 60Hz refresh. The infrared 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.

Head-Mounted Displays

Most Head-Mounted Displays (HMD) systems present the rich 3-D cues of head-motion parallax and stereopsis. They are 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 or Ascension) for precise location of the user's head in real time.

Figure 4 shows the use of an HMD.

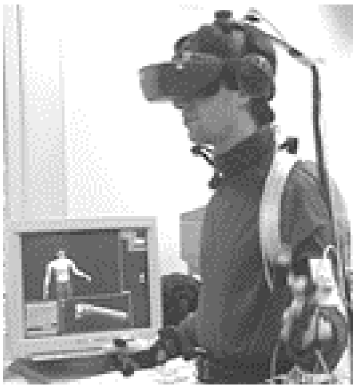


Figure 4. Head-Mounted Display

An optics model is required to specify the computation necessary to create orthostereoscopically correct images for an HMD and indicates the parameters of that system that need to be measured and incorporated into the model. To achieve orthostereoscopy, the nonlinear optical distortion must be corrected by remapping all the pixels on the screen with a predistortion function. Linear graphics primitives such as lines and polygons are written into a virtual screen image buffer, and then all the pixels are shifted according to the predistortion function and written to the screen image buffer for display. The predistortion function is the inverse of the field distortion function for the optics, so that the virtual image seen by the eye matches the image in the virtual screen buffer. A straight line in the virtual image buffer is predistorted into a curved line on the display screen, which is distorted by the optics into a line that is seen as straight.

CAVE

The CAVE(TM) is a multi-person, room-sized, high-resolution, 3D video and audio environment. It was developed at University of Illinois and is available commercially through Pyramid Systems Inc.

Currently, four projectors are used to throw full-color, computer-generated images onto three walls and the floor (the software could support a 6 wall CAVE.) CAVE software synchronizes all the devices and calculates the correct perspective for each wall. In the current configuration, one Rack Onyx with 2 Infinite Reality Engine Pipes is used to create imagery for the four walls.

In the CAVE all perspectives are calculated from the point of view of the user. A head tracker provides information about the user's position. Offset images are calculated for each eye. To experience the stereo effect, the user wears active stereo glasses which alternately block the left and right eye.

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 needed in VR, images should be digitized at the traditional video frame rate. One of the possibilities for doing this is the SIRIUS® Video card from Silicon Graphics. With SIRIUS®, images are digitized at a frequency of 25 Hz (PAL) or 30 Hz (NTSC) and may be analyzed by the VR program.

Real-time audio input

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

Speech synthesis facilities are of clear utility in a VR environment especially for command feedback. Although speech synthesis software is available even at the personal computer level, some improvement is still needed, particularly in the quality of speech.

A considerable amount of work has also been done in the field of voice recognition systems, and now commercial systems are available. But they are still expensive especially systems which are person and accent independent. Moreover, systems require a training process to go through for each user. Also, the user must be careful to leave a noticeable gap between each word which is unnatural.

2.2 Haptic interfaces and tactile feedback for VE applications

Recent developments of VE applications have enhanced the problem of user's interaction with virtual entities. Manipulation procedures consist in grasping objects and moving them among the fingers according to sequences of movements that provide a finite displacement of the grasped object with respect to the palm. Then the realistic control of the above procedures in VE implies that the man-machine interface system be capable of recording the movements of the human hand (fingers movements and gross movements of the hand) and also of replicating, on the human hand, virtual forces and contact conditions occurring when contact is detected between the virtual hand and the virtual object. Therefore hand movement recording and contact-force replication represent the two main functionalities of the interface system. At present, although several examples of tracking systems and glove-like advanced interfaces are available for hand and finger movements recording, the design of force and tactile feedback systems still presents methodological as well as technological problems. If we consider for example, the grasping of a cup, there are two main consequences:

- the VR user can reach out and grasp a cup but will not feel the sensation of touching the cup
- there is nothing to prevent the grasp continuing right through the surface of the cup!

Providing a **tactile feedback** means to provide some feedback through the skin. This may be done in gloves by incorporating vibrating nodules under the surface of the glove. This is what is available in the CyberTouch® of Virtual Technologies. CyberTouch® (Figure 5). gives a tactile feedback by featuring small vibrotactile stimulators on each finger and the palm of the CyberGlove®. Each stimulator can be individually programmed to vary the strength of touch sensation. The array of stimulators can generate simple sensations such as pulses or sustained vibration, and they can be used in combination to produce complex tactile feedback patterns. Software developers can design their own actuation profile to achieve the desired tactile sensation, including the perception of touching a solid object in a simulated virtual world. This is not a realistic simulation of touch, but it at least provides *some* indication of surface contact.

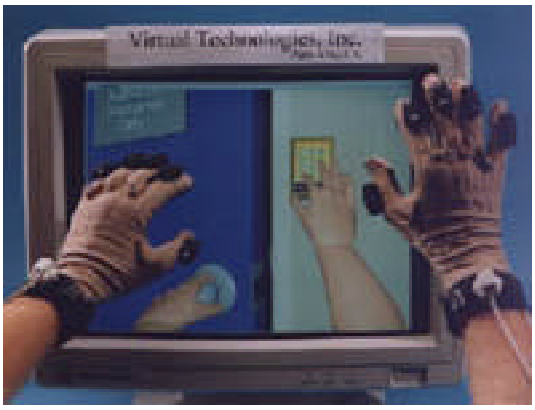


Figure 5. Use of CyberTouch®

Exos has also incorporated a tactile feedback device (Touchmaster®) into their Dextrous Hand Master. It is based on a low cost voice-coil oscillator. Another approach includes inflatable bubbles in the glove, materials that can change from liquid to solid state under electric charge and memory metals. The Teletact® Glove provides low resolution tactile feedback through the use of 30 inflatable air pockets in the glove.

Providing a means to enforce physical constraints, also simulating forces that can occur in teleoperation environments. Some devices have been built to provide **force feedback**.

The Laparoscopic Impulse Engine is a 3-D human interface specifically designed for VR simulations of Laparoscopic and Endoscopic surgical procedures. It allows a user to wield actual surgical tools and manipulated them as if performing real surgical procedures. The device allows the computer to track the delicate motions of the virtual surgical instruments while also allowing the computer to command realistic virtual forces to the user's hand. The net result is a human-computer interface which can create VR simulations of medical procedures which not only look real, but actually feel real!

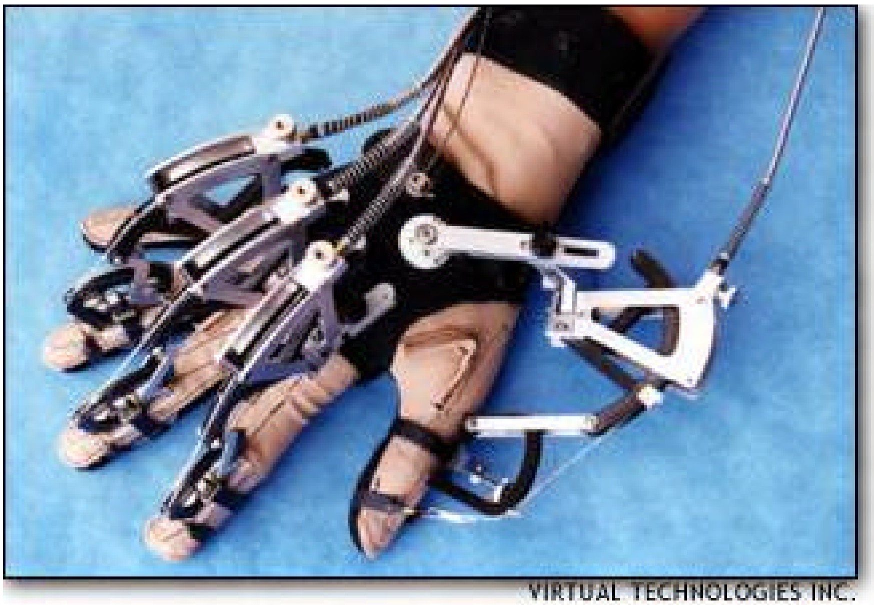
The Impulse Engine 2000 is a force feedback joystick which accurately tracks motion in two degrees of freedom and applies high fidelity force feedback sensations through the joystick handle. The Impulse Engine 2000 can realistically simulate the feel of surfaces, textures, springs, liquids, gravitational fields, bouncing balls, biological material, or any other physical sensation that you can represent mathematically. The Impulse Engine is a research quality force feedback interface with very low inertia, very low friction, and very high bandwidth.

The PHANToM® device's design allows the user to interact with the computer by inserting his or her finger into a thimble. For more sophisticated applications, multiple fingers may be used simultaneously or other devices such as a stylus or tool handle may be substituted for the thimble. The PHANToM® device provides 3 degrees of freedom for force feedback, and optionally, 3 additional degrees of freedom for measurement.

Robotic and Magnetic Interface for VR Force Interactions made by Iowa State University. It is a haptic interface system that allows force interactions with computer-generated VR graphical displays. This system is based on the application of electromagnetic principles to couple the human hand with a robotic manipulator. Using this approach, the forces are transmitted between the robot exoskeleton and the human without using mechanical attachments to the robot.

The Freedom-7® by McGill University Center for Intelligent Machines has a work area sufficient to enable a user to manipulate a tool using wrist and finger motions. Primarily intended to support the simulation of a variety of basic surgical instruments including, knives, forceps, scissors, and micro-scissors. The device incorporates a mechanical interface which enables the interchange of handles, for example to emulate these four categories of instruments, while providing the force feedback needed to simulate the interaction of an instrument with a tissue.

One of the extensions of the popular CyberGlove® that is used to measure the position and movement of the fingers and wrist is a CyberGrasp® (Figure 6). It is a haptic feedback interface that enables to actually "touch" computer-generated objects and experience force feedback via the human hand. The CyberGrasp® is a lightweight, unencumbering force-reflecting exoskeleton that fits over a CyberGlove® and adds resistive force feedback to each finger. With the CyberGrasp® force feedback system, users are able to explore the physical properties of computer-generated 3D objects they manipulate in a simulated 'virtual world.' The grasp forces are exerted via a network of tendons that are routed to the fingertips via an exoskeleton, and can be programmed to prevent the user's fingers from penetrating or crushing a virtual object. The tendon sheaths are specifically designed for low compressibility and low friction. The actuators are high-quality DC motors located in a small enclosure on the desktop. There are five actuators, one for each finger. The device exerts grasp forces that are roughly perpendicular to the fingertips throughout the range of motion, and forces can be specified individually. The CyberGrasp system allows full range-of-motion of the hand and does not obstruct the wearer's movements. The device is fully adjustable and designed to fit a wide variety of hands.



VIRTUAL TECHNOLOGIES INC.

Figure 6. CyberGrasp

The similar mechanical glove called Hand Force Feedback (HFF) was developped by Bergamasco [5] at PERCRO. They also develop a complete glove device, able to sensorize the 20 degrees of freedom of a human hand. The same laboratory developped External Force Feedback (EFF) system that is a design and realization of an arm exoskeleton. The arm exoskeleton is a mechanical structure wrapping up the whole arm of the user. The mechanical structure possesses 7 degrees of freedom corresponding to the joints of the human arm from shoulder to the wrist, and allows natural mobility to the human arm. It allows for simulation of collisions against the objects of the VE as well as the weight of "heavy" virtual objects.

We should also mention the work of several other researchers. Robinett [6] 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 [7] 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. [8] 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.

2.3 Audiospace and auditory systems

The use of sound is reported to be a surprisingly powerful cue in VR. At the minimum, binaural sound can be used to provide additional feedback to the user for such activities as grasping objects and navigation. People may easily locate the direction of a sound source. In the horizontal plane, it is based the time between the sound arriving at one ear and the other. But location of sound direction is also a learned skill. We may place small microphones in each ear and make a stereo recording that, when replayed, will recreate the feeling of directionalized sound. However, the problem in VR is that we want the position of the sound source to be independent of the user's head movement! We would like to attach recorded, live or computer generated sound to objects in the VE.

There was several attempts to solve this problem. Scott Foster at the NASA Ames VIEW Lab developed a device called the Convolvotron, which can process four independent point sound sources simultaneously, compensating for any head movement on the fly. Crystal River Engineering later developed the Maxitron, that can handle 8 sound sources as well as simulating the acoustics including sound reflection of a moderately sized room. Focal Point produce a low cost 3D audio card for Pcs and Macintoshes.

The PSFC, or Pioneer Sound Field Control System, is a DSP-driven hemispherical 14-loudspeaker array, installed at the University of Aizu Multimedia Center. Collocated with a large screen rear-projection stereographic display, the PSFC features realtime control of virtual room characteristics and direction of two separate sound sources, smoothly steering them around a configurable soundscape. The PSFC controls an entire sound field, including sound direction, virtual distance, and simulated environment (reverb level, room size and liveness) for each source.

We should also mention the work of Blauert [9] at Bochum University in Germany.

3 VR systems

3.1 Architecture of a VR system

A VR application is very often composed of a group of processes communicating through inter-process communication (IPC). As in the Decoupled Simulation Model [10], each of the processes is continuously running, producing and consuming asynchronous messages to perform its task. A central application process manages the model of the virtual world, and simulates its evolution in response to events coming from the processes that are responsible for reading the input device sensors at specified frequencies. 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.

The application process is by far the most complex component of the system. This process has to respond to asynchronous events by making the virtual world's model evolve from one coherent state to the next and by triggering appropriate visual and audio feedback. During interaction, the user is the source of a flow of information propagating from input device sensors to manipulated models. Multiple mediators can be interposed between sensors and models in order to transform the information accordingly to interaction metaphors.

3.2 Dynamics Model

In order to obtain animated and interactive behavior, the system has to update its state in response to changes initiated by sensors attached to asynchronous input devices such as timers or trackers. The application can be viewed as a network of interrelated objects whose behavior is specified by the actions taken in response to changes in the objects on which they depend.

In order to provide a maintenance mechanism that is both general enough to allow the specification of general dependencies between objects and efficient enough to be used in highly responsive interactive systems, system's state and behavior may be modeled using different primitive elements:

- *active variables*
- *hierarchical constraints*
- *daemons*

Active variables are the primitive elements used to store the system state. An active variable maintains its value and keeps track of its state changes. Upon request, an active variable can also maintain the history of its past values. This model makes it possible to elegantly express time-dependent behavior by creating constraints or daemons that refer to past values of active variables.

Multi-way relations between active variables are generally specified through **hierarchical constraints**, as introduced in ThingLab II [11]. To support local propagation, constraint objects are composed of a declarative part defining the type of relation that has to be maintained and the set of constrained variables, as well as of an imperative part, the list of possible methods that could be selected by the constraint solver to maintain the constraint.

Daemons are objects which permit the definition of sequencing between system states. Daemons register themselves with a set of active variables and are activated each time their value changes. The action taken by a daemon can be a procedure of any complexity that may create new objects, perform input/output operations, change active variables' values, manipulate the constraint graph, or activate and deactivate other daemons. The execution of a daemon's action is sequential and each manipulation of the constraint graph advances the global system time.

3.3 Dynamics and Interaction

Animated and interactive behavior can be thought of together as the fundamental problem of dynamic graphics: how to modify graphical output in response to input? Time-varying behavior is obtained by mapping dynamically changing values, representing data coming from input devices or animation scripts, to variables in the virtual world's model. The definition of this mapping is crucial for interactive applications, because it defines the way users communicate with the computer. Ideally interactive 3D systems should allow users to interact with synthetic worlds in the same way they interact with the real world, thus making the interaction task more natural and reducing training.

Mapping sensor measurements to actions

In most typical interactive applications, users spend a large part of their time entering information, and several types of input devices, such as 3D mouses and DataGloves, are used to let them interact with the virtual world. Using these devices, the user has to provide at high speed a complex flow of information, and a mapping has to be devised between the information coming from the sensors attached to the devices and the actions in the virtual world. Most of the time, this mapping is hard coded and directly dependent on the physical structure of the device used (for example, by associating different actions to the various mouse buttons). This kind of behavior may be obtained by attaching constraints directly relating the sensors' active variables to variables in the dynamic model. The beginning of the direct manipulation of a model is determined by the activation of a constraint between input sensor variables and some of the active variables in the interface of the model. While the interaction constraint remains active, the user can manipulate the model through the provided metaphor. The deactivation of the interaction constraint terminates the direct manipulation.

Such a direct mapping between the device and the dynamic model is straightforward to choose for tasks where the relations between the user's motions and the desired effect in the virtual world is mostly physical, as in the example of grabbing an object and moving it, but needs to be very carefully thought out for tasks where user's motions are intended to carry out a meaning. Adaptive pattern recognition can be used to overcome these problems, by letting the definition of the mapping between sensor measurements and actions in the virtual world be more complex, and therefore increasing the expressive power of the devices. Furthermore, the possibility of specifying this mapping through examples makes applications easier to adapt to the preferences of new users, and thus simpler to use.

Hand gesture recognition

Whole-hand input is emerging as a research topic in itself, and some sort of posture or gesture recognition is now being used in many VR systems [12]. The gesture recognition system has to classify movements and configurations of the hand in different categories on the basis of previously seen examples. Once the gesture is classified, parametric information for that gesture can be extracted from the way it was performed, and an action in the virtual world can be executed. In this way, with a single gesture both categorical and parametric information can be provided at the same time in a natural way. A visual and an audio feedback on the type of gesture recognized and on the actions executed are usually provided in applications to help the user understand system's behavior.

Gesture recognition is generally subdivided into two main portions: posture recognition, and path recognition. The posture recognition subsystem is continuously running and is responsible for classifying the user's finger configurations. Once a configuration has been recognized, the hand data is accumulated as long as the hand remains in the same posture. The history mechanism of active variables is used to automatically perform this accumulation. Data are then passed to the path recognition subsystem to classify the path. A gesture is therefore defined as the path of the hand while the hand fingers remain stable in a recognized posture. The type of gesture chosen is compatible with Buxton's suggestion [13] of using physical tension as a natural criterion for segmenting primitive interactions: the user, starting from a relaxed state, begins a primitive interaction by tensing some muscles and raising its state of attentiveness, performs the interaction, and then relaxes the muscles. In our case, the beginning of an interaction is indicated by positioning the hand in a recognizable posture, and the end of the interaction by relaxing the fingers. One of the main advantages of this technique is that, since postures are static, the learning process can be done interactively by putting the hand in the right position and indicating when to sample to the computer. Once postures are learnt, the paths can be similarly learnt in an interactive way, using the posture classifier to correctly segment the input when generating the examples. Many types of classifiers could be used for the learning and recognition task. For example in VB2 [14], feature vectors are extracted from the raw sensor data, and multi-layer perceptron networks [15] are used to approximate the functions that map these vectors to their respective classes.

Body gesture recognition

Most gesture recognition systems are limited to a specific set of body parts like hands, arms or facial expressions. However when projecting a real participant into a virtual world to interact with the synthetic inhabitants, it would be more convenient and intuitive to use body-oriented actions.

To date, basically two techniques exist to capture the human body posture in real-time. One uses video cameras which deliver either conventional or infrared pictures. This technique has been successfully used in the ALIVE system [16] to capture the user's image. The image is used for both the projection of the participant into the VE and the extraction of Cartesian information of various body parts. If this system benefits from being wireless, it suffers from visibility constraints relative to the camera and a strong performance dependence on the vision module for information extraction.

The second technique is based on magnetic sensors which are attached to the user. Most common are sensors measuring the intensity of a magnetic field generated at a reference point. The motion of the different segments is tracked using magnetic sensors (Figure 7). These sensors return raw data (e.g. positions and orientations) expressed in a single frame system. In order to match the virtual human hierarchy, we need to compute the global position of the hierarchy and the angle values of the joints attached to the tracked segments. For this purpose, an anatomical converter [17] derives the angle values from the sensor’s information to set joints of a fixed topology hierarchy (the virtual human skeleton). The converter has three important stages: skeleton calibration, sensor calibration and real-time conversion.

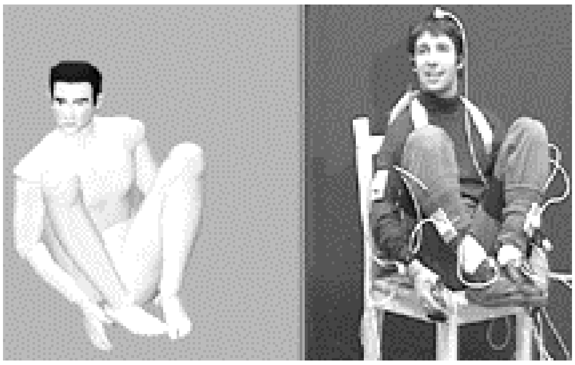


Figure 7. Tracking motion

Emering et al. [18]describe a hierarchical model of human actions based on fine-grained primitives. An associated recognition algorithm allows on-the-fly identification of simultaneous actions. By analyzing human actions, it is possible to detect three important characteristics which inform us about the specification granularity needed for the action model. First, an action does not necessarily involve the whole body but may be performed with a set of body parts only. Second, multiple actions can be performed in parallel if they use non-intersecting sets of body parts. Finally a human action can already be identified by observing strategic body locations rather than skeleton joint movements. Based on these observations, a top-down refinement paradigm appears to be appropriate for the action model. The specification grain varies from coarse at the top level to very specialized at the lowest level. The number of levels in the hierarchy is related to the feature information used. At the lowest level, the authors use the skeleton degrees of freedom (DOF) which are the most precise feature information available (30-100 for a typical human model). At higher levels, they take advantage of strategic body locations like the center of mass and end effectors, i.e. hands, feet, the head and the spine root.

Virtual Tools

Virtual tools are first class objects, like the widgets of UGA [19], which encapsulate a visual appearance and a behavior to control and display information about application objects. The visual appearance of a tool must provide information about its behavior and offer visual semantic feedback to the user during manipulation. The user declares the desire to manipulate an object with a tool by binding a model to a tool. When a tool is bound, the user can manipulate the model using it, until he decides to unbind it. When binding a model to a tool, the tool must first determine if it can manipulate the given model, identifying on the model the set of public active variables requested to activate its binding constraints. Once the binding constraints are activated, the model is ready to be manipulated. The binding constraints being generally bi-directional, the tool is always forced to reflect the information present in the model even if it is modified by other objects. Unbinding a model from a tool detaches it from the object it controls. The effect is to deactivate the binding constraints in order to suppress dependencies between tool's and model's active variables. Once the model is unbound, further manipulation of the tool will have no effect on the model. Figure 8 shows an example of the use of a SCALE tool.

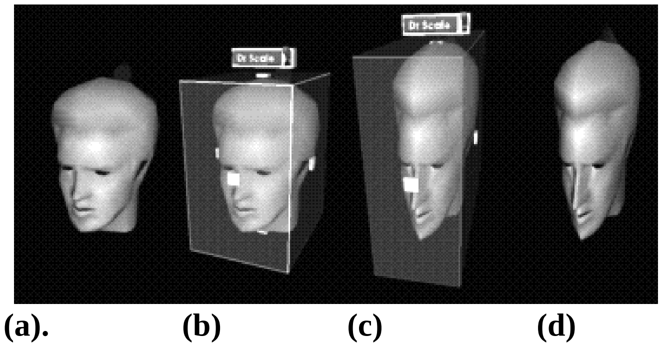


Figure 8a. Model before manipulation
b. A SCALE tool is made visible and bound to the model
c. The model is manipulated via the SCALE tool
d. The SCALE tool is unbound and made invisible

3.4 A few VR toolkits

WorldToolkit®

WorldToolkit®, developed by Sense8 Corporation, provides a complete VE development environment to the application developer. The structure of WorldToolKit® is in an object-oriented manner. The WorldToolKit® API currently consists of over 1000 high-level C functions, and is organized into over 20 classes including the universe (which manages the simulation, and contains all objects), geometrical objects, viewpoints, sensors, paths, lights, and others. Functions exist for device instancing, display setup, collision detection, loading object geometry from file, dynamic geometry creation, specifying object behavior, and controlling rendering.

WorldToolkit® uses the single loop simulation model, which sequentially reads sensors, updates the world model, and generates the images. Geometric objects are the basic elements of a universe. They can be organized in a hierarchical fashion and interact with each other. They may be stationary objects or exhibit dynamic behaviour. WorldToolKit® also provides a 'level of detail' process which corresponds to a method of creating less complex objects from the detailed object.

Each universe is a separate entity and can have different rules or dynamic behaviour imposed on its objects. Moving between different universes in WorldToolKit® is achieved by portals, which are assigned to specific polygons. When the user's viewpoint crosses the designated polygon the adjacent universe is entered.The idea of a portal is rather like walking through a door into another room. With this approach, it is possible to create several smaller universes together to make one large VE.

MR Toolkit

MR (Minimal Reality) Toolkit was developed by researchers at University of Alberta [20]. The MR Toolkit is in the form of a subroutine library that supports the development of VR applications. The toolkit supports various tracking devices, distribution of the user interface and data to multiple workstations, real-time performance interaction and analysis tools.

The MR toolkit is comprised of three levels of software. At the lowest level is a set of device-dependent *packages*. Each package consists of a client/server software pair. The server is a process that continuously samples the input device and performs further processing such as filtering; while the client is a set of library routines that interface with the server. The second, middle, layer consists of functions that convert the ‘raw’ data from the devices to the format more convenient for the user interface programmer. Additionally, routines such as data transfer among workstations and work space mapping reside in this layer. The top layer consists of high level functions that are used for average VE interface. For example, a single function to initialize all the devices exists in this layer. Additionally, this layer contains routines to handle synchronization of data and operations among the workstations.

Other three-dimensional toolkits

Other Toolkits, such as IRIS Performer from Silicon Graphics Inc., Java3D, OpenGL Optimizer, etc. also support the development of VR applications, however they are low-level libraries for manipulation of the environment, viewpoints, display parameters. They do not address support for I/O devices, participant representation, motion systems and networking. Therefore, they do not address rapid prototyping of NVE applications. Consequently, we regard these toolkits as instruments to develop VEs, rather than architectures.

4 Virtual Humans in Virtual Environments

The participant should animate his virtual human representation in real-time, however the human control is not straightforward: the complexity of virtual human representation needs a large number of degrees of freedom to be tracked. In addition, interaction with the environment increases this difficulty even more. Therefore, the human control should use higher level mechanisms to be able to animate the representation with maximal facility and minimal input. We can divide the virtual humans according to the methods to control them:

- Directly controlled virtual humans
- User-guided virtual humans
- Autonomous virtual humans
- Interactive Perceptive Actors

4.1 Direct controlled virtual humans

A complete representation of the participant's virtual body should have the same movements as the real participant body for more immersive interaction. This can be best achieved by using a large number of sensors to track every degree of freedom in the real body. Molet et al. [17] discuss that a minimum of 14 sensors are required to manage a biomechanically correct posture, and Semwal et al. [21] present a closed-form algorithm to approximate the body using up to 10 sensors. However, many of the current VE systems use head and hand tracking. Therefore, the limited tracking information should be connected with human model information and different motion generators in order to "extrapolate" the joints of the body which are not tracked. This is more than a simple inverse kinematics problem, because there are generally multiple solutions for the joint angles to reach to the same position, and the most realistic posture should be selected. In addition, the joint constraints should be considered for setting the joint angles.

4.2 Guided virtual humans

Guided virtual humans are those which are driven by the user but which do not correspond directly to the user motion. They are based on the concept of real-time direct metaphor [22], a method consisting of recording input data from a VR device in real-time allowing us to produce effects of different natures but corresponding to the input data. There is no analysis of the real meaning of the input data. The participant uses the input devices to update the transformation of the eye position of the virtual human. This local control is used by computing the incremental change in the eye position, and estimating the rotation and velocity of the body center. The walking motor uses the instantaneous velocity of motion, to compute the walking cycle length and time, by which it computes the joint angles of the whole body. The sensor information or walking can be obtained from various types of input devices such as special gesture with DataGlove, or SpaceBall, as well as other input methods.

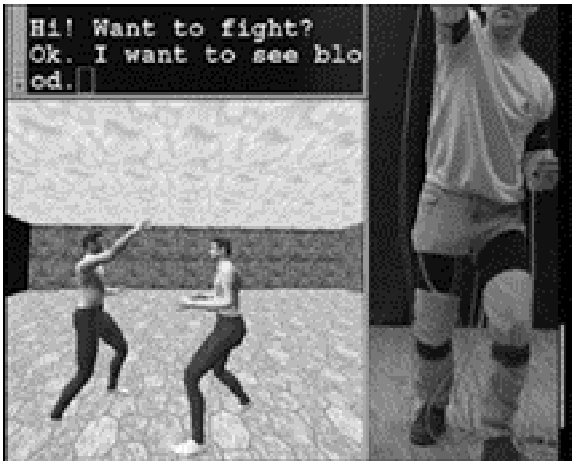
4.3 Autonomous virtual humans

Autonomous actors are able to have a behavior, which means they must have a manner of conducting themselves. The virtual human is assumed to have an internal state which is built by its goals and sensor information from the environment, and the participant modifies this state by defining high level motivations, and state changes Typically, the actor should perceive the objects and the other actors in the environment through virtual sensors [23]: visual, tactile and auditory sensors. Based on the perceived information, the actor’s behavioral mechanism will determine the actions he will perform. An actor may simply evolve in his environment or he may interact with this environment or even communicate with other actors. In this latter case, we will consider the actor as a interactive perceptive actor.

The concept of virtual vision was first introduced by Renault et al. [24] as a main information channel between the environment and the virtual actor. The synthetic actor perceives his environment from a small window in which the environment is rendered from his point of view. As he can access z-buffer values of the pixels, the color of the pixels and his own position, he can locate visible objects in his 3D environment. To recreate the virtual audition [25], it requires a model a sound environment where the Virtual Human can directly access to positional and semantic sound source information of a audible sound event. For virtual tactile sensors, our approach [26] is based on spherical multi-sensors attached to the articulated figure. A sensor is activated for any collision with other objects. These sensors have been integrated in a general methodology for automatic grasping.

4.4 **Interactive Perceptive Actors**:

We define an interactive perceptive synthetic actor [27] as an actor aware of other actors and real people. Such an actor is also assumed to be autonomous of course. Moreover, he is able to communicate interactively with the other actors whatever their type and the real people. For example, Emering et al. describe how a directly controlled Virtual Human performs fight gestures which are recognized by a autonomous virtual opponent [18] as shown in Figure 9. The latter responds by playing back a pre-recorded keyframe sequence.



Hi! Want to fight?
Ok. I want to see blo
od.

Figure 9. Fight between a participant and an interactive perceptive actor

4.5 Facial communication in Virtual Environments

For the representation of facial expressions in Networked VEs, four methods are possible: video-texturing of the face, model-based coding of facial expressions, lip movement synthesis from speech and predefined expressions or animations.

Video-texturing of the face

In this approach the video sequence of the user's face is continuously texture mapped on the face of the virtual human. The user must be in front of the camera, in such a position that the camera captures his head and shoulders, possibly together with the rest of the body. A simple and fast image analysis algorithm is used to find the bounding box of the user's face within the image. The algorithm requires that head and shoulder view is provided and that the background is static (though not necessarily uniform). Thus the algorithm primarily consists of comparing each image with the original image of the background. Since the background is static, any change in the image is caused by the presence of the user, so it is fairly easy to detect his/her position. This allows the user a reasonably free movement in front of the camera without the facial image being lost.

Model-based coding of facial expressions

Instead of transmitting whole facial images as in the previous approach, in this approach the images are analyzed and a set of parameters describing the facial expression is extracted**.** As in the previous approach, the user has to be in front of the camera that digitizes the video images of head-and-shoulders type. Accurate recognition and analysis of facial expressions from video sequence requires detailed measurements of facial features. Recognition of the facial features may be primarily based on color sample identification and edge detection [28]. Based on the characteristics of human face, variations of these methods are used in order to find the optimal adaptation for the particular case of each facial feature. Figure 10 illustrates this method with a sequence of original images of the user (with overlaid recognition indicators) and the corresponding images of the synthesized face.

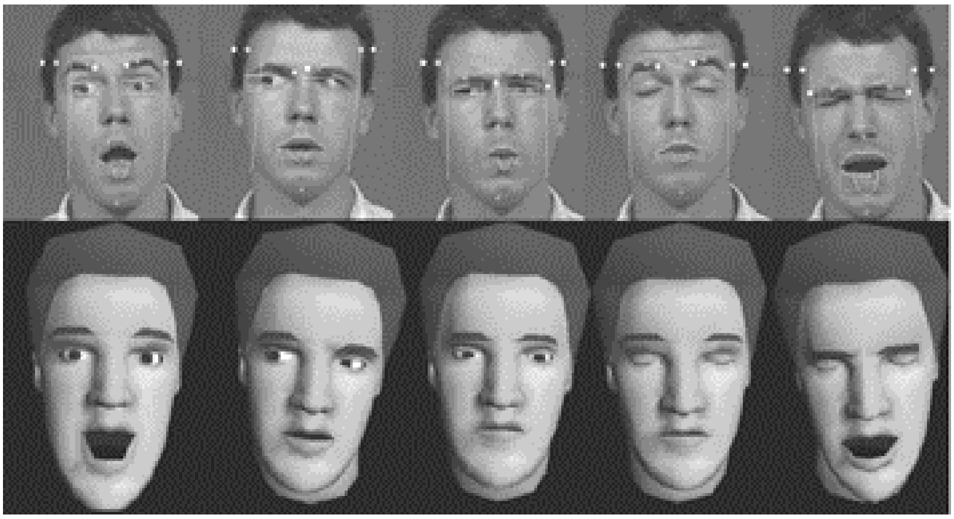


Figure 10. Model-based coding of the face

Lip movement synthesis from speech

It might not always be practical for the user to be in front of the camera (e.g. if he doesn't have one, or if he wants to use an HMD). Lavagetto [29] shows that it is possible to extract visual parameters of the lip movement by analyzing the audio signal of the speech.

Predefined expressions or animations

In this approach the user can simply choose between a set of predefined facial expressions or movements (animations). The choice can be done from the keyboard through a set of "smileys" similar to the ones used in e-mail messages.

5 Networked Virtual Environments

5.1 Introduction

Networking coupled with highly interactive technology of virtual worlds will dominate the world of computers and information technology. It will not be enough to produce slick single-user, standalone virtual worlds. Networked VE (NVE) systems will have to connect people, systems, information streams and technologies with one another. The information that is currently shared through file systems or through other "static" media will have to be exchanged through the network. This information has to reside "in the net" where it is easy to get at. Developing VEs that support collaboration among a group of users is a complex and time-consuming task. In order to develop such VEs, the developer has to be proficient in network programming, object management, graphics programming, device handling, and user interface design. Even after gaining expertise in such diverse specializations, developing network-based VEs takes a long time since network-based programs are inherently more difficult to program and debug than standalone programs.

Providing a behavioral realism is a significant requirement for systems that are based on human collaboration, such as Computer Supported Cooperative Work (CSCW) systems. Networked CSCW systems also require that the shared environment should: provide a comfortable interface for gestural communication, support awareness of other users in the environment, provide mechanisms for different modes of interaction (synchronous vs. asynchronous, allowing to work in different times in the same environment), supply mechanisms for customized tools for data visualization, protection and sharing. VR can provide a powerful mechanism for networked CSCW systems, by its nature of emphasizing the presence of the users in the VE.

Until recently, networked graphics applications were prototype systems, demonstrating the effectiveness of the technology. However, a current effort is to provide real applications, manifested by the 3D graphics interchange standardization efforts such as VRML 2.0 and MPEG-4. The main contributors in these standards are from the industry, that expect to diffuse their application content using these standards.

There have been an increasing interest in the area of NVEs [30] [31] [32] recently and in the next Sections, we will describe the most important systems.

5.2 State-of-the-art in NVEs

VEOS

VEOS (Virtual Environment Operating Shell), developed by the University of Washington, was one of the first complete NVE architectures to provide an integrated software to develop general applications. VEOS [33] uses tightly-integrated computing model for management of data, processes, and communication in the operating system level, hiding details from the applications as much as possible.

dVS

dVS, developed by Division Ltd in UK [34], is one of the commonly used VE commercial development tools available today. The system aims to provide a modular line for creating and interacting with virtual prototypes of CAD products. The architecture is based on dividing the environment into a number of autonomous entities, and processing them in parallel. It is designed to suit a range of different parallel architectures. It supports loosely coupled networks, symmetric multiprocessors and single processor systems. An entity represents high level 3D objects, which encapsulate all the elements of the object.

DIVE

DIVE (Distributed Interactive Virtual Environment) [35] [36] is developed at the Swedish Institute of Computer Science. The DIVE system is a toolkit for building distributed VR applications in a heterogeneous network environment. The networking is based on reliable multicast communication, using the ISIS Toolkit [37]. DIVE uses peer-to-peer communication to implement shared VEs. The DIVE run-time environment consists of a set of communicating processes, running on nodes distributed within a local area network (LAN) or wide area network (WAN). The processes, representing either human users or autonomous applications, have access to a number of databases, which they update concurrently. Each database contains a number of abstract descriptions of graphical objects that together constitute a VE. Associated with each world is a process group, consisting of all processes that are members of that world. Multicast protocols are used for the communication within such a process group [38].

NPSNET

NPSNET has been created at the Naval Postgraduate School in Monterey by Zyda et al. [39]. It uses an object- and event-based approach to distributed, interactive virtual worlds for battlefield simulation and training. Virtual worlds consist of objects that interact with each other by broadcasting a series of events. An object initiating an event does not calculate which other objects might be affected by it. It is the receiving object's responsibility to determine whether the event is of its interest or not. To minimize communication processing and bandwidth requirements, objects transmit only changes in their behavior. Until an update is received, the new position of a remote object is extrapolated from the states last reported by those objects.

NPSNET can be used to simulate an air, ground, nautical (surface or submersible) or virtual vehicle, as well as human subjects. The standard user interface devices for navigation include a flight control system (throttle and stick), a SpaceBall®, and/or a keyboard. The system models movement on the surface of the earth (land or sea), below the surface of the sea and in the atmosphere. Other entities in the simulation are controlled by users on other workstations, who can either be human participants, rule-based autonomous entities, or entities with scripted behavior. The VE is populated not only by users' vehicles/bodies, but also by other static and dynamic objects that can produce movements and audio/visual effects. NPSNET succeeds to provide an efficient large-scale networked VE using general-purpose networks and computers and the standard communication protocol, DIS.

MASSIVE

MASSIVE (Model, Architecture and System for Spatial Interaction in Virtual Environments) [40] [41] was developed at the University of Nottingham. The main goals of MASSIVE are scalability and heterogeneity, i.e. supporting interaction between users whose equipment has different capabilities and who therefore employ radically different styles of user interface, e.g. users on text terminals interacting with users wearing Head Mounted Displays and magnetic trackers.

MASSIVE supports multiple virtual worlds connected via portals. Each world may be inhabited by many concurrent users who can interact over ad-hoc combinations of graphics, audio and text interfaces. The graphics interface renders objects visible in a 3D space and allows users to navigate this space with six degrees of freedom. The audio interface allows users to hear objects and supports both real-time conversation and playback of preprogrammed sounds. The text interface provides a plan view of the world via a window (or map) that looks down onto a 2D plane across which users move (similar to Multi-User Dungeons).

SPLINE

SPLINE (Scaleable PLatform for Interactive Environments), developed by Mitsubishi Electric Research Labs is a software platform that allows to create virtual worlds featuring: multiple, simultaneous, geographically separated users; multiple computer simulations interacting with the users; spoken interaction between the users; immersion in a 3D visual and audio environment; and comprehensive run-time modifiability and extendibility. The system’s main application theme is social VR, where people interact using their embodiments. An important feature of SPLINE is the support for both pre-recorded and real-time audio.

The MERL group, developers of SPLINE [42], have created an application, called *Diamond Park*. The park consists of a square mile of detailed terrain with visual, audio, and physical interaction. The participants navigate around the scene through bicycling, using an exercise bike as physical input device; and their embodiment moves on a virtual bicycle with speed calculated from the force exerted on the physical bicycle.

BRICKNET

BRICKNET [43], developed at ISS (Institute of System Sciences, Singapore), is designed for the creation of virtual worlds that operate on workstations connected over a network and share information with each other, forming a loosely coupled system. The BRICKNET toolkit provides functionalities geared towards enabling faster and easier creation of networked virtual worlds. It eliminates the need for the developer to learn about low level graphics, device handling and network programming by providing higher level support for graphical, behavioral and network modeling of virtual worlds. BRICKNET introduces an object sharing strategy which sets it apart from the classic NVE mindset. Instead of all users sharing the same virtual world, in BRICKNET each user controls his/her own virtual world with a set of objects of his/her choice. He/she can then expose these objects to the others and share them, or choose to keep them private. The user can request to share other users' objects providing they are exposed. So, rather than a single shared environment, BRICKNET is a set of "overlapping" user-owned environments that share certain segments as negotiated between the users.

VISTEL

Ohya et al. [44] from ATR Research Lab in Japan propose VISTEL (Virtual Space teleconferencing system). As the name indicates, the purpose of this system is to extend teleconferencing functionality into a virtual space where the participants can not only talk to each other and see each other, but collaborate in a 3D environment, sharing 3D objects to enhance their collaboration possibilities. The current system supports only two users and does not attempt to solve problems of network topology, space structuring or session. The human body motion is extracted using a set of magnetic sensors placed on the user's body. Thus the limb movements can be captured and transmitted to the receiving end where they are visualized using an articulated 3D body representation. The facial expressions are captured by tracking facial feature points in the video signal obtained from a camera.

VLNET

Virtual Life Network (VLNET) [45] [46] is a general-purpose client/server NVE system using highly realistic virtual humans for user representation. VLNET achieves great versatility through its open architecture with a set of interfaces allowing external applications to control the system functionality.

Figure 11 presents a simplified overview of the architecture of a VLNET client. The VLNET core performs all the basic system tasks: networking, rendering, visual data base management, user management including body movement and facial expressions. A body deformation module is integrated in the client core. When actors are animated, each client updates the skin shapes of all visible virtual actors within the client's field of view. A set of simple shared memory interfaces is provided through which external applications can control VLNET. The VLNET drivers also use these interfaces. The drivers are small service applications provided as part of VLNET system that can be used to solve some standard tasks, e.g. generate walking motion, support navigation devices like mouse, SpaceBall, etc. The connection of drivers and external applications to VLNET is established dynamically at runtime based on the VLNET command line.

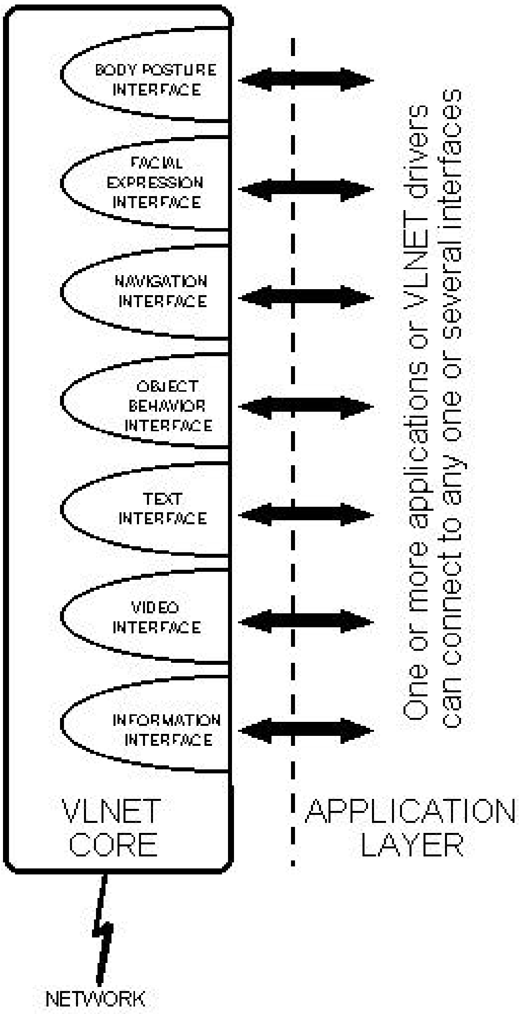

BODY POSTURE INTERFACE
FACIAL EXPRESSION INTERFACE
NAVIGATION INTERFACE
OBJECT BEHAVIOR INTERFACE
TEXT INTERFACE
VIDEO INTERFACE
INFORMATION INTERFACE
VLNET CORE
APPLICATION LAYER
One or more applications or VLNET drivers can connect to any one or several interfaces
NETWORK

Figure 11. Simplified view of VLNET client architecture

The *Facial Expression Interface* is used to control expressions of the user's face. The expressions are defined using the Minimal Perceptible Actions (MPAs) [47]. The MPAs provide a complete set of basic facial actions. By using them it is possible to define any facial expression. The *Body Posture Interface* controls the motion of the user's body. The postures are defined using a set of joint angles corresponding to 72 degrees of freedom of the skeleton model [48] used in VLNET. The *Navigation Interface* is used for navigation, hand and head movement, basic object manipulation and basic system control. All movements are expressed using matrices. The basic manipulation includes picking objects up, carrying them and letting them go, as well as grouping and ungrouping of objects. The system control provides access to some system functions that are usually accessed by keystrokes, e.g. changing drawing modes, toggling texturing, displaying statistics. The *Object Behavior Interface* is used to control the behavior of objects. Currently it is limited to the controlling of motion and scaling, defined by matrices passed to the interface. It is also used to handle the sound objects; i.e. objects that have prerecorded sounds attached to them. The Object Behavior Interface can be used to trigger these sounds. The *Video Interface* is used to stream video texture (as well as static textures) onto any object in the environment. The Alpha channel can be used for blending and achieving effects of mixing real and virtual objects/persons. The interface accepts requests containing the image (bitmap) and the ID of an object on which the image is to be mapped. The image is distributed and mapped on the requested object at all sites. The *Text Interface* is used to send and receive text messages to and from other users. An inquiry can be made through the text interface to check if there are any messages, and the messages can be read. The interface gives the ID of the sender for each received message. A message sent through the text interface is passed to all other users in a VLNET session. The *Information Interface* is used by external applications to gather information about the environment from VLNET. It provides high-level information while isolating the external application from the VLNET implementation details. It also allows two ways of obtaining information, namely the request-and-reply mechanism and the event mechanism.

Figure 12 shows a VLNET session for interactive tennis playing in a shared environment [49].

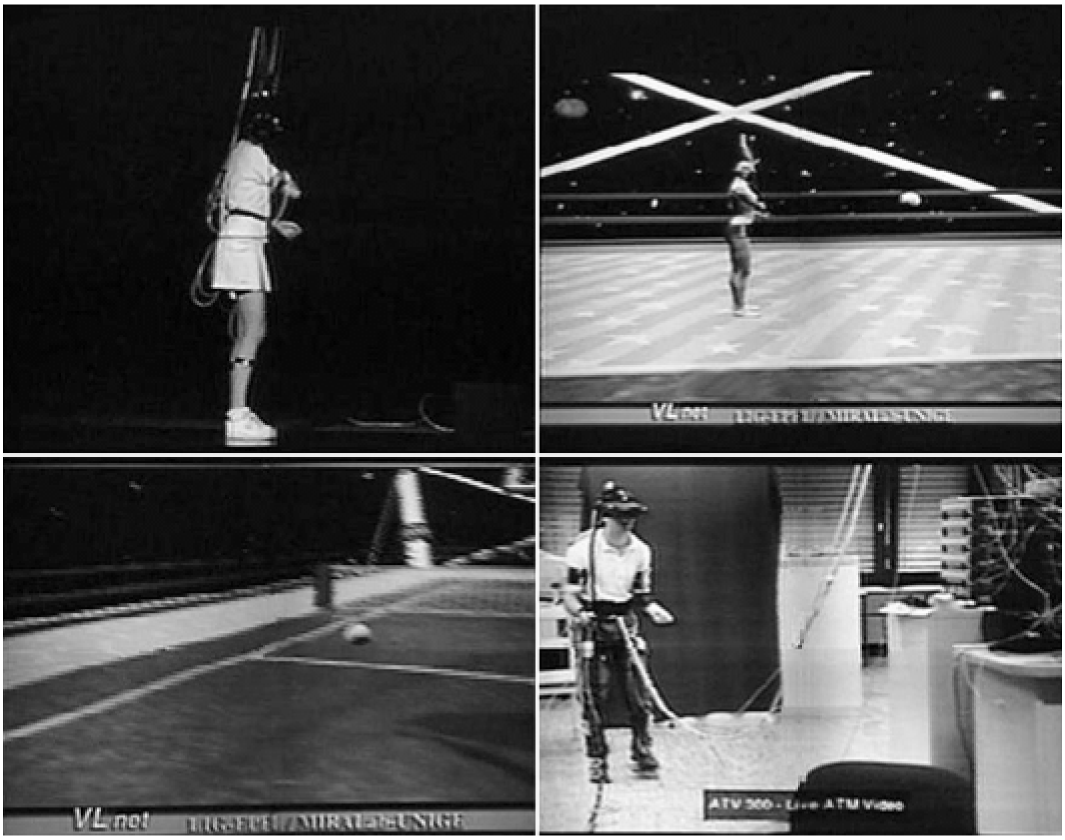


Figure 12. Anyone for Tennis : a VLNET session

6 Applications of Virtual Reality

VR may offer enormous benefits to many different applications areas. This is one main reason why it has attracted so much interest. VR is currently used to explore and manipulate experimental data in ways that were not possible before.

Operations in dangerous environments

There are still many examples of people working in dangerous or hardship environments that could benefit from the use of VR-mediated teleoperation. Workers in radioactive, space, or toxic environments could be relocated to the safety of a VR environment where they could 'handle' any hazardous materials without any real danger using teleoperation or telepresence.

Moreover, the operator's display can be augmented with important sensor information, warnings and suggested procedures. However, teleoperation will be really useful when further developments in haptic feedback will come.

Scientific visualization

Scientific Visualization provides the researcher with immediate graphical feedback during the course of the computations and gives him/her the ability to 'steer' the solution process. Similarly, by closely coupling the computation and visualization processes, Scientific Visualization provides an exploratory, experimentation environment that allows the investigators to concentrate their efforts on the important areas. VR could bring a lot to Scientific Visualization by helping to interpret the masses of data.

A typical example of Scientific Visualization is the NASA Virtual Wind Tunnel at the NASA Ames Research Center. In this application, the computational fluid dynamicist controls the computation of virtual smoke streams emanating from his/her fingertips. Another application at NASA Ames Research Center is the Virtual Planetary Exploration. It helps planetary geologists to remotely analyze the surface of a planet. They use VR techniques to roam planetary terrains using complex height fields derived from Viking images of Mars.

Medicine

Until now experimental research and education in medicine was mainly based on dissection and study of plastic models. Computerized 3D human models provide a new approach to research and education in medicine. Experimenting medical research with virtual patients will be a reality. We will be able to create not only realistic looking virtual patients, but also histological and bone structures. With the simulation of the entire physiology of the human body, the effects of various illnesses or organ replacement will be visible. Virtual humans associated with VR will certainly become one of the medical research tools of the next century.

One of the most promising application is surgery. The surgeon using an HMD and DataGloves may have a complete simulated view, including his/her hands, of the surgery. The patient should be completely reconstructed in the VE, this requires a very complete graphics human database. For medical students learning how to operate, the best way would be to start with 3D virtual patients and explore virtually all the capabilities of surgery.

By modeling deformation of human muscles and skin, we will gain fundamental insight into these mechanisms from a purely geometric point of view. This has promise of application, for example, in the pathology of skin repair after burning.

One other important medical application of virtual humans is orthopedics. Once a motion is planned for a virtual human, it should be possible to alter or modify a joint and see the impact on the motion.

Rehabilitation and help to disable people

It is also possible to create dialogue based on hand gestures [50] such as a dialogue between a **deaf** real human and a **deaf** virtual human using American Sign Language. The real human signs using two DataGloves, and the coordinates are transmitted to the computer. Then a sign-language recognition program interprets these coordinates in order to recognize gestures. A dialogue coordination program then generates an answer or a new sentence. The sentences are then translated into the hand signs and given to a hand animation program which generates the appropriate hand positions.

We may also think about using VR techniques to improve the situation of disabled patients after brain injuries. VR may play a supportive role in memory deficiencies, impaired visual-motor performance or reduced vigilance.

Muscular dystrophy patients can learn to use a wheelchair through VR.

Psychiatry

Another aspect addressed by Whalley [51] is the use of VR and virtual humans in psychotherapies. Whalley states that VR remains largely at the prototype stage – images are cartoon-like and carry little conviction. However, with the advent of realistic virtual humans, it will be possible to recreate situations in a Virtual World, immersing the real patient into virtual scenes, for example, to re-unite the patient with a deceased parent, or to simulate the patient as a child allowing him or her to re-live situations with familiar surroundings and people.

With a VR-based system, it will be also possible in the future to change parameters for simulating some specific behavioral troubles in psychiatry. Therapists may also use VR to treat sufferers of child abuse and people who are afraid of heights.

Architectural visualization

In this area, VR allows the future customer to “live” in his/her a new house before it is built. He/she could get a feel for the space, experiment with different lighting schemes, furnishings, or even the layout of the house itself. A VR architectural environment *can* provide that feeling of space. Once better HMDs become available, VR design environment will be a serious competitive advantage.

Design

Many areas of design are typically 3D as for example, the design of a car shape, where the designer looks for sweeping curves and good aesthetics from every possible view. Today's design tools are mouse or stylus/digitizer based and thereby force the designer to work with 2D input devices. For many designers, this is difficult since it forces them to mentally reconstruct the 3D shape from 2D sections. A VR design environment can give to designers appropriate 3D tools.

Education and training

VR promises many applications in simulation and training. The most common example is the flight simulator. This type of simulator has shown the benefits of simulation environments for training. They have lower operating costs and are safer to use than real aircraft. They also allow the simulation of dangerous scenarios not allowable with real aircraft. The main problem of current flight simulators is that they cannot be used for another type of training like submarine training for example.

Simulation and ergonomy

VR is a very powerful tool to simulate new situations especially to test the efficiency and the ergonomy. For example, we may produce immersive simulation of airports, train stations, metro stations, hospitals, work places, assembly lines, pilot cabins, cockpits, access to control panel in vehicles and machines. In this area, the use of Virtual humans is essential and even simulation of crowds [52] is essential. We may also mention game and sport simulation.

Computer supported cooperative work

Shared VR environment can also provide additional support for cooperative work. They allow possibly remote workers to collaborate on tasks. However, this type of system requires very high bandwidth networks like ATM connecting locations and offices. However, it surely saves time and money for organizations. Network VR simulations could enable people in many different locations to participate together in teleconferences, virtual surgical operations, teleshopping (Figure 13), or simulated military training exercises.

Entertainment

This is the area which starts to drive the development of VR technology. The biggest limiting factor in VR research today is the sheer expense of the technology. It is expensive because the volumes are low. For entertainment, mass production is required. Another alternative is the development of "Virtual Worlds" for Lunaparks/casinos.

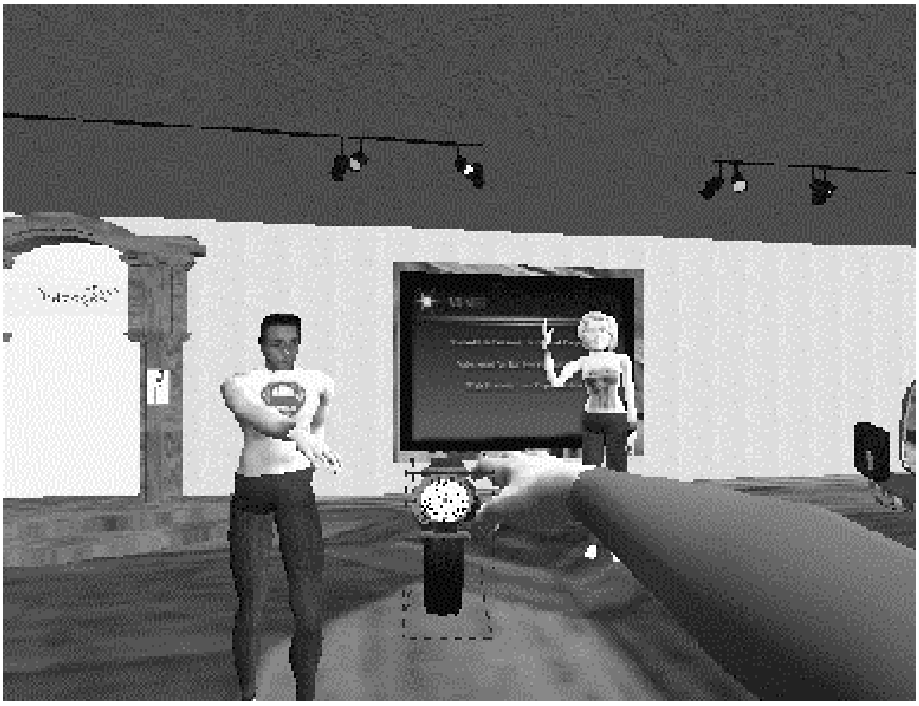


Figure 13. Collaborative Virtual Presentation Application (using VLNET)

7 References

[1] Ellis SR (1991) Nature and Origin of Virtual Environments: A Bibliographic Essay, Computing Systems in Engineering, 2(4), pp.321-347.

[2] Astheimer P, Dai, Göbel M, Kruse R, Müller S, Zachmann G (1994) Realism in Virtual Reality, in: Magnenat Thalmann N and thalmann D, Artificial Life and Virtual reality, John Wiley, pp.189-209.

[3] Slater M, Usoh M (1994) Body Centred Interaction in Immersive Virtual Environments, in: Magnenat Thalmann N and thalmann D, Artificial Life and Virtual reality, John Wiley, pp.125-147.

[4] Ware C, Jessome DR (1988) Using the Bat: a six-dimensional mouse for object placement, IEEE CG&A Vol 8(6) pp 65-70 (1988).

[5] Bergamasco M (1994) Manipulation and Exploration of Virtual Objects, in: Magnenat Thalmann N and Thalmann D, Artificial Life and Virtual Reality, John Wiley, pp.149-160.

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

[7] 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.

[8] 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.

[9] Blauert J (1983) Spatial Hearing, The Psychophysics of Human Sound Localization, MIT Press, Cambridge.

[10] Shaw C, Liang J, Green M, Sun Y (1992), The Decoupled Simulation Model for Virtual Reality Systems. Proc. SIGCHI, pp.321-328.

[11] Borning A, Duisberg R, Freeman-Benson B, Kramer A, Woolf M (1987), Constraint Hierarchies, Proc. OOPSLA:, pp.48-60.

[12] Sturman DJ (1991), Whole-Hand Input, PhD Thesis, MIT.

13] Buxton WAS (1990), A Three-state model of Graphical Input. In Diaper D, Gilmore D, Cockton G, Shackel B (Editors) Human-Computer Interaction: Interact, Proceedings of the IFIP Third International Conference on Human-Computer Interaction, North-Holland, Oxford.

[14] Gobbetti E, Balaguer JF, Thalmann D (1993) VB2: An Architecture For InteractionIn Synthetic Worlds, Proc. UIST ’93, ACM.

[15] Rumelhart DE, Hinton GE, Williams RJ (1986), Learning Internal Representations by Error Propagation. In Rumelhart DE, McClelland JL (Editors) Parallel Distributed Processing, Vol. 1: 318-362.

[16] Maes P, Darrell T, Blumberg B, Pentland A (1995) The ALIVE system: Full-body interaction with Autonomous Agents, Proceedings of the Computer Animation'95 Conference, Geneva, Switzerland, IEEE-Press.

[17] Molet T, Boulic R, Thalmann D (1996) A Real-Time Anatomical Converter for Human Motion Capture, Proc. 7h Eurographics Workshop on Animation and Simulation, Springer-Verlag, WiWare ...en, September 1996.

[18] Emering L, Boulic R, Thalmann D, Interacting with Virtual Humans through Body Actions, IEEE Computer Graphics and Applications, 1998 , Vol.18, No1, pp8-11.

[19] Conner DB, Snibbe SS, Herndon KP, Robbins DC, Zeleznik RC, Van Dam A (1992), Three-Dimensional Widgets. SIGGRAPH Symposium on Interactive 3D Graphics: 183-188.

[20] Shaw C, Green M (1993) The MR Toolkit Peers Package and Experiment, Proc. IEEE Virtual Reality Annual International Symposium, pp 463-469.

[21] Semwal SK, Hightower R, Stansfield S (1996) Closed Form and Geometric Algorithms for Real-Time Control of an Avatar, Proc. VRAIS 96, pp.177-184.

[22] Thalmann D (1993) Using Virtual Reality Techniques in the Animation Process in: Virtual Reality Systems (Earnshaw R, Gigante M, Jones H eds), Academic Press, pp.143-159.

[23] Thalmann D (1995) Virtual Sensors: A Key Tool for the Artificial Life of Virtual Actors, Proc. Pacific Graphics ‘95, Seoul, Korea, 1995, pp.22-40.

[24 Renault O, Magnenat Thalmann N, Thalmann D, A Vision-based Approach to Behavioural Animation, The Journal of Visualization and Computer Animation, Vol 1, No 1, 1990, pp.18-21.

[25] Noser H, Thalmann D (1995) Synthetic Vision and Audition for Digital Actors, Proc.Eurographics‘95, 1995, pp.325-336.

[26] Huang Z, Boulic R, Magnenat Thalmann N, Thalmann D (1995) A Multi-sensor Approach for Grasping and 3D Interaction, Proc. CGI ‘95, Academic Press, pp.235-254.

[27] Thalmann D (1996) A New Generation of Synthetic Actors: the Interactive Perceptive Actors, Proc. Pacific Graphics ‘96 Taipeh, Taiwan, 1996, pp.200-219.

[28] Pandzic I, Kalra P, Magnenat Thalmann N, Thalmann D (1994) Real Time Facial Interaction, Displays, Vol.15, No3, Butterworth, pp.157-163.

[29] Lavagetto F (1995) Converting Speech into Lip Movements: A Multimedia Telephone for Hard of Hearing People, IEEE Trans. on Rehabilitation Engineering, Vol.3, N1, pp.90-102.

[30] Zeltzer D., Johnson M., Virtual Actors and Virtual Environments, Interacting with Virtual Environments, MacDonald L., Vince J. (Ed), 1994.

[31] Stansfield S., "A Distributed Virtual Reality Simulation System for Simulational Training", Presence: Teleoperators and Virtual Environments, Vol. 3, No. 4, 1994.

[32] Gisi MA, Sacchi C (1994) Co-CAD: A Collaborative Mechanical CAD System, Presence: Teleoperators and Virtual Environments, Vol. 3, No. 4.

[33] Bricken W, Coco G (1993) The VEOS Project, Technical Report R-93-3, Human Interface Technology Laboratory, University of Washington.

[34] Grimsdale C (1991) dVS - Distributed Virtual environment System, Proc. Computer Graphics '91 Conference, London, Blenheim Online ISBN 0 86353 282 9.

[35] Carlsson C Hagsand O (1993) DIVE - a Multi-User Virtual Reality System, Proc. IEEE Virtual Reality Annual International Symposium (VRAIS'93), Sept. 18-22, 1993, Seattle, Washington, USA, pp.394-400.

[36] Fahlen LE, Stahl O, Brown CG, Carlsson C (1993). A space-based model for user- interaction in shared synthetic environments, Proc. ACM InterCHI'93, Amsterdam, Holland, 24-29 April 1993, pp:43-48.

[37] Birman K, Cooper R, Gleeson B (1991) Programming with process groups: Group and multicast semantics, Technical Report TR-91-1185, Dpt CS, Cornell University

[38] Birman K (1991) Maintaining Consistency in Distributed Systems, Technical report TR91-1240, Dpt CS, Cornell Uniuversity, 1991

[39] Zyda MJ, Pratt DR, Monahan JG, Wilson KP (1992) NPSNET: Constructing a 3D Virtual World, Proc. 1992 Symposium on Interactive 3D Graphics, 29 March - 1 April 1992, pp.147-156.

[40] Benford S, Bowers J, Fahlen LE, Greenhalgh C, Mariani J, Rodden T (1995) Networked Virtual Reality and Cooperative Work, Presence: Teleoperators and Virtual Environments, Vol.4, No.4, pp.364-386

[41] Greenhalgh C, Benford S (1995) MASSIVE, A Distributed Virtual Reality System Incorporating Spatial Trading, Proc. the 15th International Conference on Distributed Computing Systems, Los Alamitos, CA, ACM, pp 27-34.

[42] Waters RC, Anderson DB, Barrus JW, Brogan DC, Casey MC, G McKeown SG, Nitta T, Sterns IB, Yerazunis WS (1997) Diamond Park and Spline: Social Virtual Reality with 3D Animation, Spoken Interaction, and Runtime Extendability, Presence, MIT Press, Vol.6, No4, pp.461-481.

[43] Singh G, Serra L, Png W, Wong A, Ng H (1995) BrickNet: Sharing Object Behaviors on the Net, Proc. IEEE VRAIS '95, pp.19-27.

[44] Ohya J, Kitamura Y, Kishino F, Terashima N (1995) Virtual Space Teleconferencing: Real-Time Reproduction of 3D Human Images, Journal of Visual Communication and Image Representation, Vol.6, No.1, pp.1-25.

[45] Pandzic I, Magnenat Thalmann N, Capin T, Thalmann (1997) Virtual Life Network: A Body-Centered Networked Virtual Environment, Presence, MIT, Vol. 6, No 6, 1997, pp. 676-686.

[46] Capin T, Pandzic I, Magnenat Thalmann N, Thalmann D (1997) Virtual Human Representation and Communication in the VLNET Networked Virtual Environments, IEEE Computer Graphics and Applications, Vol.17, No2, 1997, pp.42-53.

[47] 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, pp.59-69.

[48] Boulic R, Capin T, Huang Z, Moccozet L, Molet T, Kalra P, Lintermann B, Magnenat-Thalmann N, Pandzic I, Saar K, Schmitt A, Shen J, Thalmann D (1995) The HUMANOID Environment for Interactive Animation of Multiple Deformable Human Characters, Proc. Eurographics '95, Maastricht, pp.337-348.

[49] Molet T, Aubel A, Çapin T, Carion S, Lee E, Magnenat Thalmann N, Noser H, Pandzic I, Sannier G, Thalmann D (1998) Anyone for Tennis, Presence (to appear)

[50] Broeckl-Fox U, Kettner L, Klingert A, Kobbelt L (1994) Using Three-Dimensional Hand-Gesture Recognition as a New 3D Input Technique, in: Magnenat Thalmann N, Thalmann D (eds) Artificial Life and Virtual Reality, John Wiley.

[51] Whalley LJ (1993) Ethical Issues in the Application of Virtual Reality to the Treatment of Mental Disorders, in: Earnshaw et al. (eds) Virtual Reality Systems, Academic Press, pp.273-288.

[52] Musse SR, Thalmann D (1997) A Model of Human Crowd Behavior, Computer Animation and Simulation '97, Proc. Eurographics workshop, Budapest, Springer Verlag, Wien, pp.39-51.