International Journal in Computer Simulation 5, 287–303 (1995)

Virtual Reality Graphics Simulation with Force Feedback

GRIGORE BURDEA
DANIEL GOMEZ
NOSHIR LANGRANA
EDWARD ROSKOS
Rutgers- The State University of New Jersey

PAUL RICHARD
University of Paris VI

Force feedback, although an important enhancement to virtual reality simulations, is not presently available in commercial systems. This article presents a prototype force feedback master integrated in a network-distributed, single-user virtual reality simulation. The experimental set-up utilizes four computers which perform force calculation, hand gesture recognition, sound generation, and graphics display. Human-factors test results show up to 70% reduction in the error rate over ten subjects when force feedback was present in the simulation. The learning time for new virtual reality tasks was also reduced by 50% versus an open-loop simulation.

force feedback, portable master, DataGlove, object model, compliance simulation, human factors

1 INTRODUCTION

Virtual reality is a computer-generated immersive environment with which users have real-time, multisensorial interactions. These interactions involve all human senses through visual feedback [1], 3-D sound [2], force and touch feedback [3,4], and even smell and taste [5]. Because of this rich information transfer virtual reality is emerging as a very powerful human-machine interface technology and will become part of high-end graphics workstations [6].

At the present time the tools needed to bring feedback signals to the user's hand are still under research [7,8]. Commercial systems such as sensing gloves used to transmit hand gestures [9] or exoskeletons [10] operate in open loop, with no feedback to the user. This is an important limitation on the realism of the computer simulation.

The research reported here was supported by the CAIP Center, Rutgers University, with funds provided by the New Jersey Commission on Science and Technology and by CAIP's industrial members, by National Science Foundation Grants MSS-89-09335 and CCR-89-09197, and by the French Ministry of Education.

Correspondence and requests for reprints should be sent to Grigore Burdea, Rutgers University, CORE Bldg, CAIP Center, Room 721, Frelinghuysen Road, P.O. Box 1390, Piscataway, NJ 08855-1390.

■ Manuscript received December 23, 1992; manuscript revised September 25, 1993.

Work on gloves that have touch and force feedback has resulted in a number of prototypes [11–13]. These systems are compact, light, and desk-top, and therefore differ from more complex systems developed for robot teleoperation [14].

Because these hardware tools are just emerging, very little data are available to quantify the influence of touch and force feedback on human performance in a virtual environment [15,16]. These data are, however, needed in the design of future feedback systems as well as the development of new applications.

Software environments for virtual reality are also under research. The difficulty of producing realistic, interactive, computer graphics in real-time has long been recognized [17]. This is because of the large number of polygons that need to be rendered every second, as well as to the computations associated with object(s) interactions. When additional feedback signals such as sound, force, or touch feedback are provided, as is the case in virtual reality simulations, then the computational and input/output loads grow even more.

A general consensus among researchers is that computing loads in virtual reality simulations need to be distributed. This can be done over the network by assigning force, sound, and taste feedback to other computers than the graphics engine [18,19]. Added concurrency can be implemented by distributing the load among multiprocessors in a unified system using parallel architectures [20].

This article presents a single-user, network-distributed, system for virtual environments with graphics, sound, and force feedback. The task studied by our simulation involves precision grasping of light virtual objects. As opposed to power grasping, where contact is with the whole palm area [21], precision grasping has contact at the fingertips only. Most human touch sensors are located at the fingertips [22], so precision grasping is commonly employed for complex object manipulation where touch information is important.

The user interacts with the objects in a virtual room. The objects, depending upon their compliance, deform and return simulated forces when grasped and manipulated [23]. The objects behave approximately as they would in the physical world, bouncing around the room while exhibiting the effects of simulated gravity. Sound provides an additional channel of information on what is happening in the virtual world.

Some of the problems investigated in this article have bearing on remote multi-user interactions where network distribution of the virtual reality simulation is unavoidable. The main thrust in this investigation is the integration of force feedback in a distributed architecture. Section 2 describes the experimental set-up and hardware characteristics. Section 3 presents some of the techniques applied in producing our virtual world and describes object models for both elastic and plastic deformations. Section 4 shows the test results of the effect of force feedback on human performance. Concluding remarks and future work directions are given in Section 5.

2 EXPERIMENTAL SET-UP

The Rutgers Portable Master with Force Feedback (or "Rutgers Master") [24] allows a user to "feel" virtual objects during tasks involving precision grasping.

The master is a compact feedback structure that fits in the palm of a sensing glove as shown in Figure 1. In the present set-up, the sensing glove is a DataGlove [9] that measures hand gestures using optical fibres and a Polhemus sensor which are mounted on the back of the hand. The optical fibres change refractance according to the bending of finger joints allowing the determination of individual joint angles. The Polhemus sensor transmits 3-D wrist position and orientation data using low-frequency magnetic fields produced by a stationary source.

The feedback structure consists of three (possible four) pneumatic microcylinders that press against the fingertips using "fork" attachments. The actuator's other extremity is attached to a small L-shaped plastic platform with sphere joints, which allow the grasping and adduction/abduction motion of each finger. The placement of the actuators in the palm avoids cables or pulleys and makes the system simpler and lighter (about 50 grams for the feedback structure). The lightness of the feedback structure is important in order to reduce operator fatigue during the simulation. Another advantage of this system is the preservation of hand freedom of motion in the virtual simulation (force feedback joysticks sacrifice this freedom of motion as the hand is kept on the desk).

The attachment of the feedback structure to the glove is done with Velcro™ strips mounted on the palm and on the fingers. These detachable connections allow for adjustments to the hand characteristics of different users.

The feedback actuators are controlled by analog proportional pressure regulators (PPR) that are housed in a master interface. These regulators control air pressure to the actuators in the palm of the user's hand. The interface has its own power supply and main air pressure indicator as well as separate LED displays for each output channel. These LEDs visualize the level of feedback forces on each of the fingers and provide an additional cue to the user and to other persons watching the simulation.

The step response for a force feedback actuator is presented in Figure 2. The transient ripple and overshoot are small enough not to be noticed by the user. A rise time of 14 ms is caused by static friction in the pneumatic cylinder and the inertia of the pressure regulator valve. The static friction is also responsible for the steady-state error of 4% of the total force of about 1 lbf/actuator. The relaxing time of 62 ms is the bottleneck for the actuator bandwidth (of 8–10 Hz). This delay is a result of the slow rate by which air is exhausted, since mufflers are installed to reduce noise.

3 IMPLEMENTATION TECHNIQUES

The virtual reality system configuration is illustrated in Figure 3. A loosely coupled, client-server architecture allows for the distribution of computation for the simulation on four workstations. A Sun4/380 is dedicated to reading and calibrating glove data, updating the level of output forces, and maintaining state information on all objects in the virtual world, while a HP755CRX workstation is dedicated to graphics rendering and display. The user may use a trackball connected to a Sun

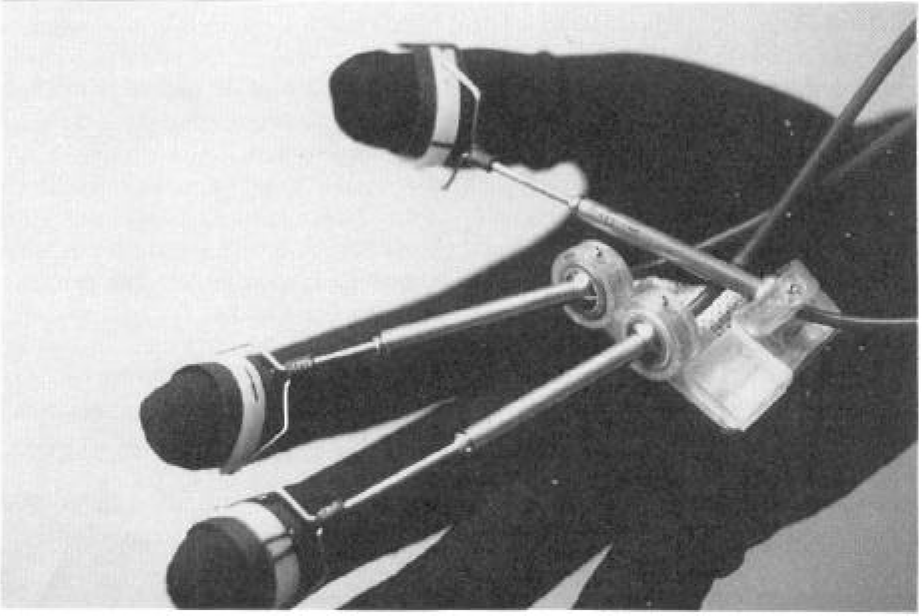


Figure 1. The Rutgers Portable Force Feedback Master.

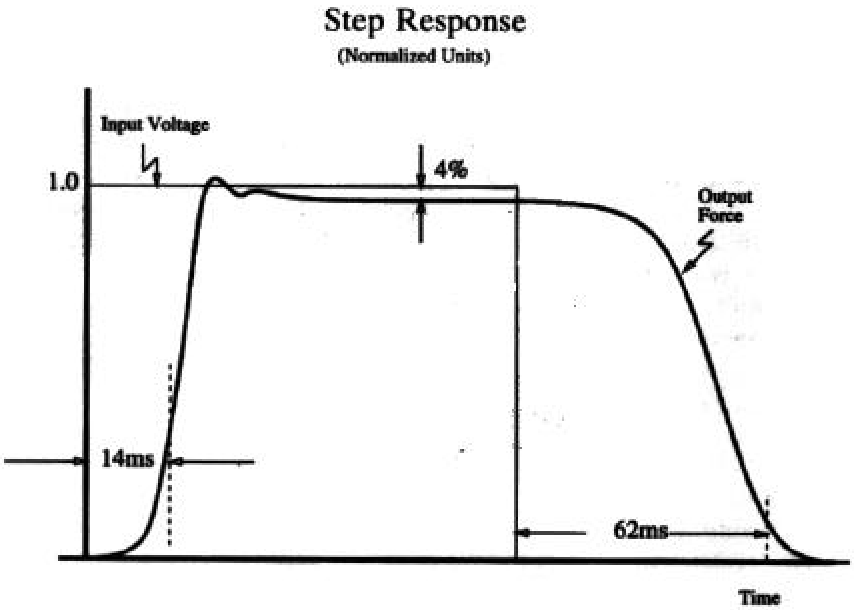


Figure 2. Response of actuators in the Rutgers Master.

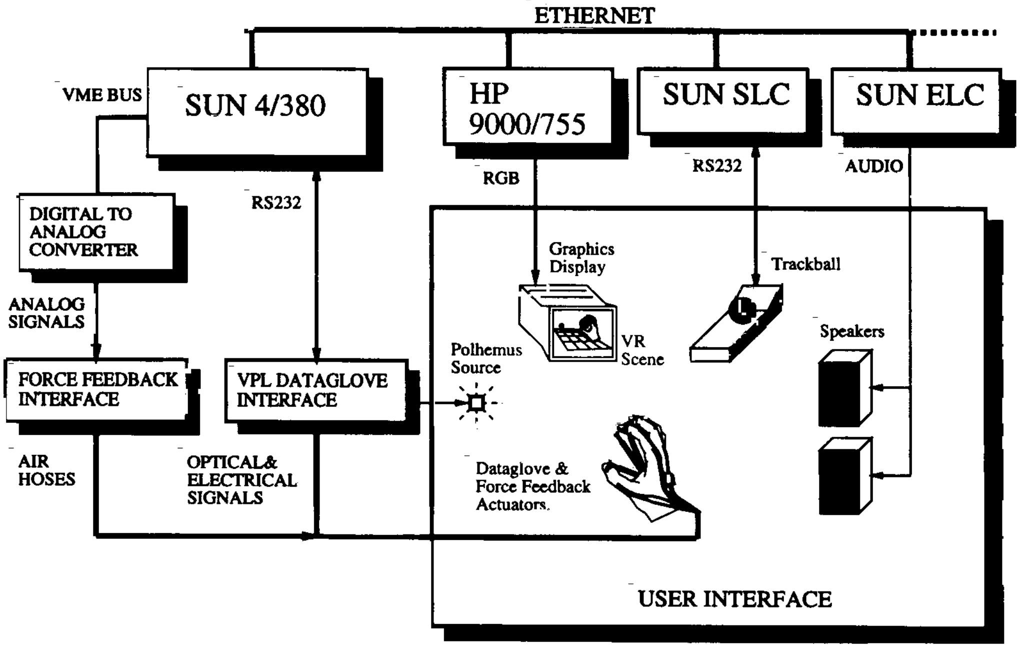


Figure 3. Virtual reality system architecture.

SLC to vary the perspective view of the virtual world. A Sun ELC is in charge of the audio display corresponding to the object interactions.

The main loop of the simulation resides on the Sun4 to which are connected the DataGlove electronic unit and the force feedback interface. The main loop reads glove data and converts them into hand orientation and position data (a calibration procedure has been developed to individualize the user's hand anatomy [25,26]. These data are then used to update the state of various objects in the simulation, taking into account object collisions, grasping, squeezing, and the "tossing" of objects. Gravity is also simulated in the world, allowing the ball to bounce around the virtual room. Update information on object location and orientation is sent by a Sun4 to a dedicated graphics workstation, currently an HP-755CRX, which runs the graphics loop (in the past, an HP-375 workstation was used, it has been upgraded because it has a refresh rate of less than 5 frames/s).

The trackball server processes raw data into view point transformations and sends these data over the Ethernet to the HP graphics renderer. When collisions between objects occur, an appropriate sound is generated. A sound ID is sent by the Sun4 to the Sun ELC running a sound server. This ID is then used by the sound server to query a database for the appropriate sound to be generated. The sound server displays the corresponding sound as soon as its ID is received. The associated networking delay is very small and is almost not perceived by the user.

The software testbed is illustrated in Figure 4. As stated, there are two loops executing independently in the simulation, the main loop and the graphics rendering loop. The main loop cycles at around 30 Hz, while graphics refresh occurs at around

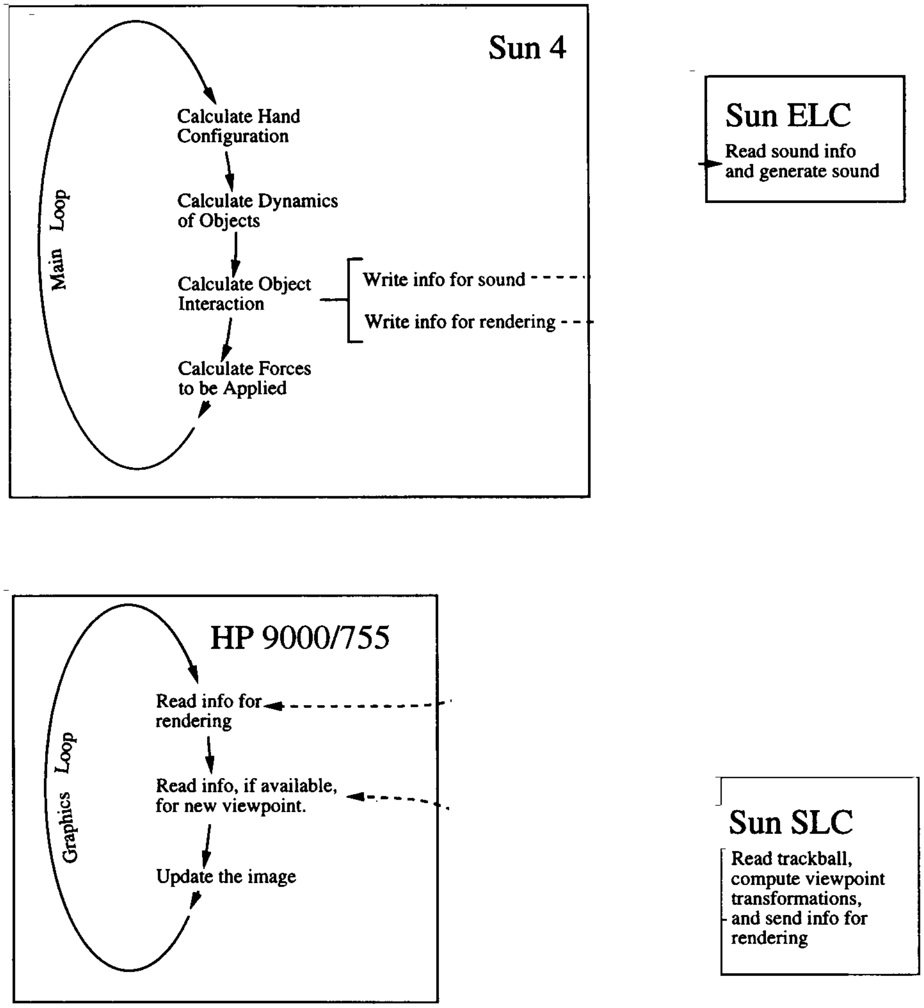


Figure 4. A view of the software testbed.

27 frames/s. The main loop will send data to the graphics renderer each cycle. Because our graphics workstation does not support interrupt-driven network I/O, network buffer sizes are made appropriately small to allow the main loop to use a `select( )` system call to determine when the next data set should be read. This way the graphics workstations does not have to query for new data and always receives data at a rate comparable to its consumption. There is, however, an added delay in ethernet networking, but this delay is relatively small.

The simulated hand has the same kinematics as the human hand, that is, four degrees of freedom per finger. The DataGlove does not measure the distal joint angles for the index, middle, ring, or small fingers. Therefore, in order to allow for normal virtual hand animation, a coupling formula was applied to determine distal joint angles based on the angle of the middle joint [23].

Three types of objects can be found in our experimental virtual world. There are walls, hands, and objects to be manipulated (currently a ball and a soda can). For the hand, ball, and can, an "×" shadow mark is drawn on the floor under the object to provide a visual cue for depth perception. The objects are programmed into a display list using the Starbase graphics library [27], double buffering, and Gouraud shading with one light source (adding more light sources will, of course, slow down the rendering).

When an object is grasped, the main loop computes the degree of object deformation based on programmed object compliance. The main loop then updates voltages on the D/A board which drives the force feedback actuators. The degree of object deformation is also provided to the graphics renderer so grasped objects appear deformed.

Object deformation has been previously studied using partial derivatives and finite element methods [28–30]. Our simulation, however, uses linearized deformation laws because of the real-time requirement of virtual reality interactions and limited computing power. When an object is grasped, the distance the object has been squeezed is used to determine the "force" to be returned to the user. Hooke's Law, $F_i = k\Delta x_i$, has been used to relate depth of compression to the generated force. In this way, the equation is kept simple enough for rapid computation while still retaining the ability to model objects of varying stiffnesses. When released, elastic objects such as a rubber ball reform to their original state, whereas plastic objects remain deformed.

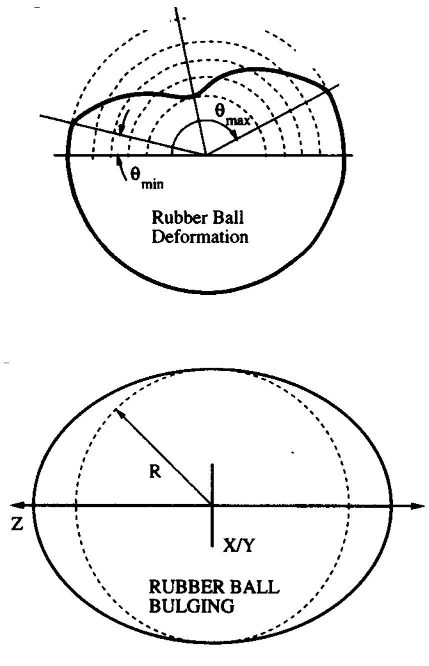
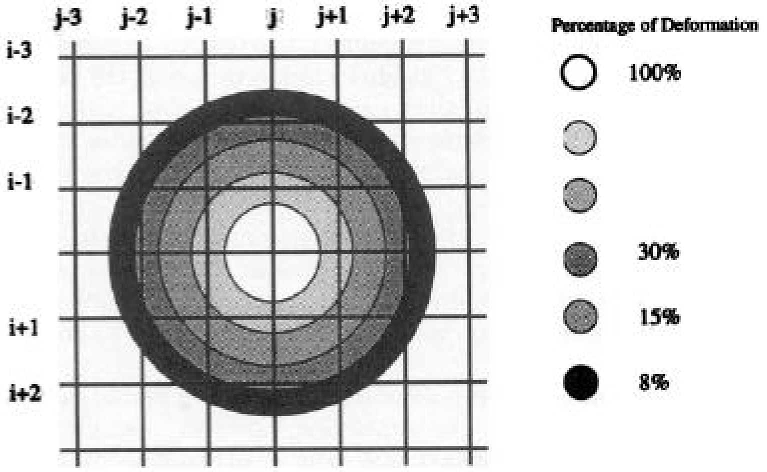
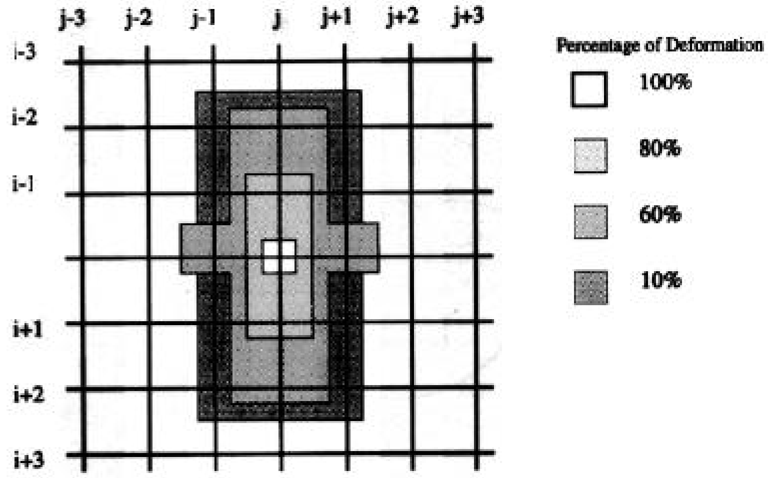


Figure 5. Sphere deformation.



RUBBER BALL MESH



SODA CAN MESH

Figure 6. Deformation regions for ball and soda can.

When the ball is deformed, the main loop sends information on the point of contact and the degree of deformation. The graphics renderer then uses these data to select a set of vertices for modification. These vertices are then mapped to spheres concentric with the ball but with smaller radii, as shown in Figure 5. The determination of the radii of the smaller spheres and the corresponding mapped vertices is outlined in Figure 6. Mapping by regions is an approximation that appears

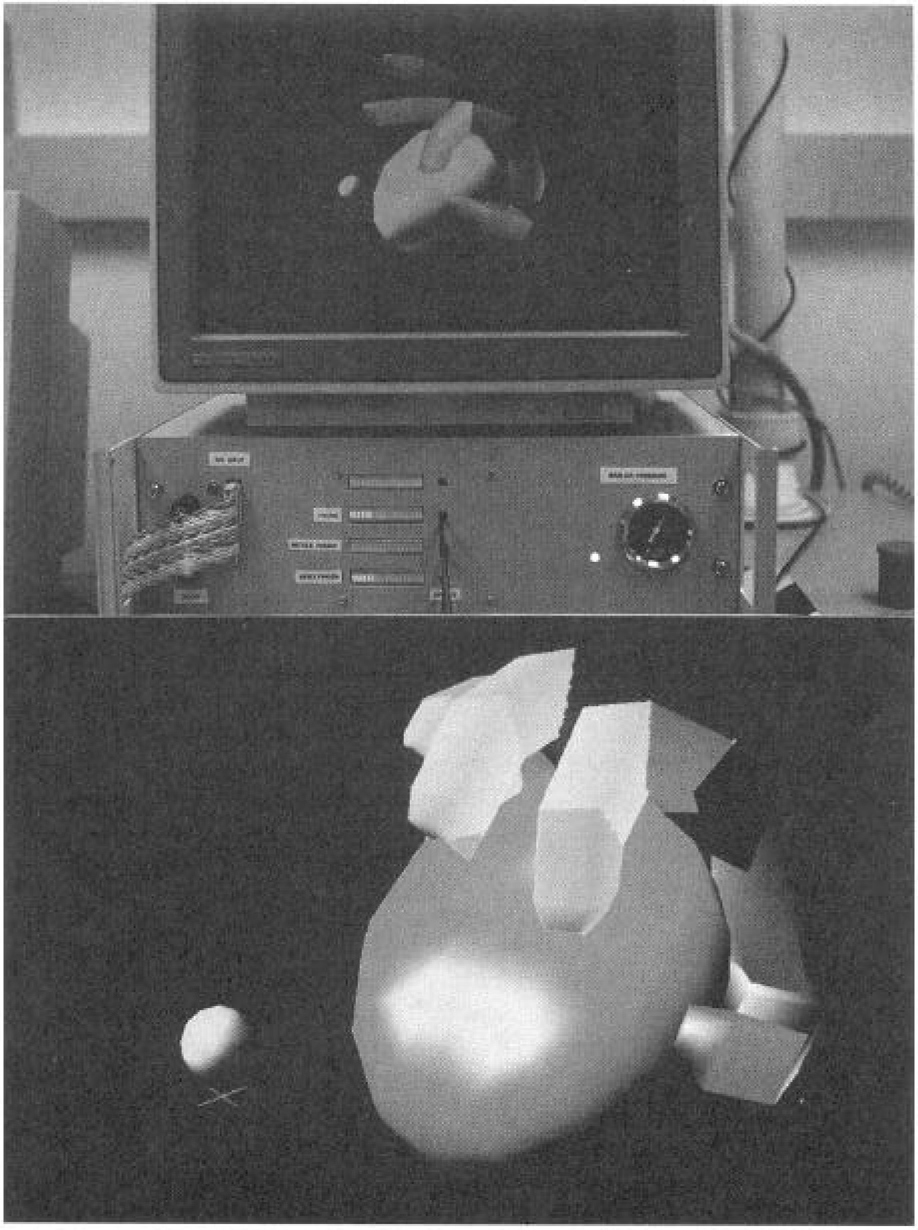


Figure 7. Hand squeezing a virtual ball.

realistic and requires little computation. A sample of the graphics output is shown in Figure 7.

To make the deformation appear more realistic, the ball "bulges" proportionally to the extent to which the ball is squeezed. The bulging effect is created by changing the z-component of each vertex as a function of *cos* θ for the entire mesh

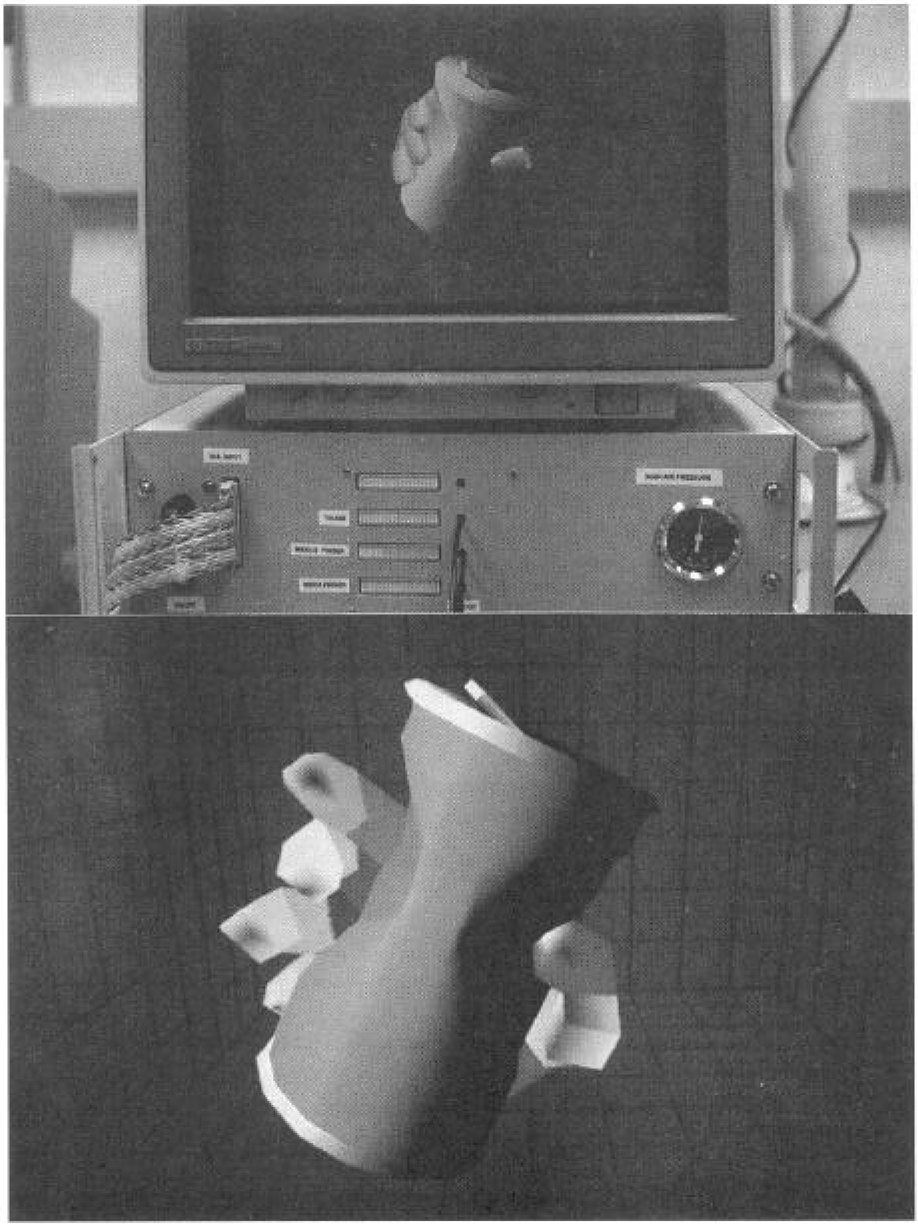


Figure 8. Hand squeezing a virtual soda can.

as shown in Figure 5. The x and y components are not changed, so the bulging effect is more noticeable at the poles than at the equator.

The soda can becomes dented when the fingertip is within the can's radial distance from its major axis. The denting algorithm also uses a mapping, but to cylinders of smaller radii, as shown in Figure 6. Whereas the rubber ball has only elastic deformations, the soda can has two deformation regions. For relatively small deformations, the can behaves elastically and reforms once released. As the can is squeezed further, the can behaves plastically and remains dented when released. Denting is discretized into states $s_0, s_1, \ldots, s_n$, where larger indices represent larger dents. When a can is dented to state s_i, it cannot enter a state s_j for $j < i$. The can cannot be squeezed beyond state s_n and so has a maximum degree of deformation. Force is present only when the fingers are in contact with the can. Figure 8 shows a sample of the graphics output for soda can deformation.

Collision detection between the hand and an object is determined by computing the position of the object in the coordinate frame of the hand using data from the Polhemus tracker. The rotation and translation matrices used are shown in Equations (1) and (2).

$$ {}^{palm}R_{global} \quad \left.\begin{matrix} C_pC_r & S_rC_p & -S_p \\ -S_rC_y + C_rS_pS_y & C_rC_y + S_rS_pS_y & C_pS_y \\ S_rS_y + C_rS_pC_y & -C_rS_y + S_rS_pC_y & C_pC_y \end{matrix}\right] = \begin{bmatrix} \vec{n} \\ \vec{s} \\ \vec{a} \end{bmatrix} \tag*{1)} $$

$$ T = \quad \begin{matrix} -xn_0 & -yn_1 & -zn_2 \\ -xs_0 & -ys_1 & -zs_2 \\ -xa_0 & -ya_1 & -za_2. \end{matrix} \tag{2} $$

Here, $S_p = \sin(pitch)$, $S_r = \sin(roll)$, and $S_y = \sin(yaw)$.

Two types of collisions may take place between the hand and the ball. In the first type, the user either slaps the ball with the palm or the back of the hand. The second type results in a grasping of the ball. Distinction between these two types of interaction is made by determining the "gesture" of the hand. A fist gesture is used to symbolize a grasping gesture, whereas an open hand indicates the user wishes to slap the ball. When the ball is slapped, it reacts as though it collided with a wall moving with the velocity of the hand. In this way, fingers do not affect the computation, but the velocity and orientation of the palm do.

4 TEST RESULTS

An earlier experimental set-up burdened the graphics machine with the tasks of computing forces, collision detection, and providing a visual force display. This earlier simulation used a less sophisticated hand model with fewer degres of freedom, bounded sphere collision detection, and did not feature object dynamics or sound. Under this load, a bandwidth of about 1.5 frames/s was achieved. This slow refresh rate was, in part, a consequence of using an older graphics workstation capable of only 2.3K polygons/s and 7.5K triangle strips/s. Current workstations are capable of over 1 million 3-D triangles/s.

To improve upon this low graphics performance, several changes were implemented. First, the visual force display was moved from software and implemented in hardware (LEDs). Then, the computations for collision detection and force determination were moved to a Sun4 workstation. With these changes, a more sophisticated hand model could be integrated into the simulation while obtaining 5 frames/s with the same older workstation. Because object management was off-loaded from the graphics machine, better collision detection and object dynamics, including the ability to slap and toss the ball, were added without cost to the graphics refresh rate. Sound was later added, again without affecting the graphics refresh rate.

We replaced the older HP workstation with a top-of-the-line HP755-CRX48 with a refresh rate of 28 frames/s. The question then became whether the 8–10 Hz force feedback bandwidth is not a limitation to the simulation. Hogan [31] estimates that the human "high-level" force compliance control loop has a very low bandwidth of about 1–2 Hz. It is this control loop that is important in the present set-up. Therefore, we do not need 30 Hz on the force feedback interface.

A number of human-factors studies were performed to determine the effect of force feedback on a user's ability to operate in the virtual world [32]. The tests involved 10 subjects (five male and five female) who were "computer literate" but had never experienced virtual reality. These tests were done with our original experimental set-up (with graphics refresh of 1.5 frames/s). The tests involved grasping virtual objects first in open loop, then in a closed loop with vision and force feedback, and finally in closed loop blindfolded. Each test was repeated 12 times and the number of errors recorded in each case.

For example, a subject had to squeeze the virtual soda can without denting it permanently. To do this, the user had to apply small enough forces to stay in the elastic deformation region. An error was recorded each time the can was plastically dented. These errors were averaged over the subject group and the standard deviation and mean were calculated. The results are shown in Figure 9. Force feedback for hard virtual objects ($k = 40$) reduced the average error rate over all subjects by more than 70%. The error rate was larger for soft objects ($k = 7$) because dead friction in the actuators masked the small feedback forces applied.

Another area of interest was the influence of force feedback on the learning time for new simulation tasks. The learning process was quantified by the drop in average error rates for repeated trials of the same task. For the soda can denting test described above, this amounted to performing the test in both open and closed loops, where the use of force feedback resulted in a 50% reduction in learning time. This is illustrated in Figure 10. These results indicate that force feedback had a positive effect on task performance and was a definite plus when compared with open-loop simulation.

These tests were subsequently repeated using the newer workstation [33]. The tests were consistent with earlier results [32] which indicates that the improvement in error rates and learning time was not due to a slow graphics display but to the presence of force feedback.

5 CONCLUSION AND FUTURE WORK

Virtual reality simulations are enhanced by the addition of force feedback to the user hand. A testbed using the Rutgers Master with three Sun4 workstations and

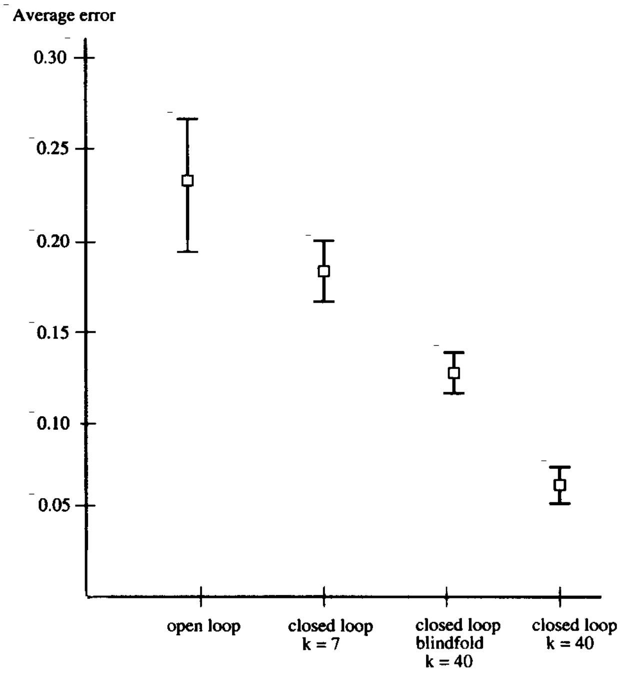


Figure 9. Comparison of error rates for various control modalities in virtual reality [32]. ©EC2 (reprinted with permission)

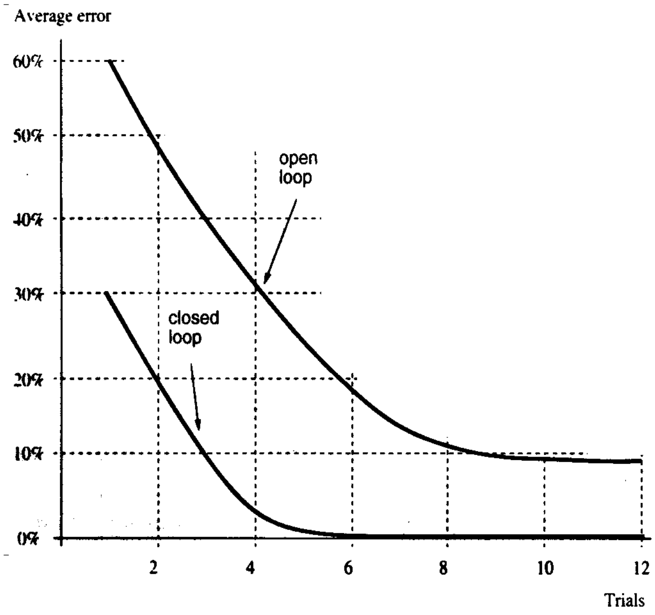


Figure 10. Learning cycles in virtual reality simulation [32]. ©EC2 (reprinted with permission)

an HP graphics workstation was presented. The processes are organized in a loosely coupled, nonscalable architecture in order to improve system performance. Two types of light virtual objects were modeled, one elastic (ball) and one plastic (soda can). The objects may be grasped and tossed around a virtual room. Whereas the graphics have been simplified because of real-time constraints, force feedback has proven to be a definite advantage for the simulation as qualified by the results of human factors tests.

An improved force feedback master is under development. This new master, called Rutgers Master II, will have more degrees of freedom, a larger work envelope, and its own position-sensing hardware, without the need for a separate sensing glove.

REFERENCES

[1] K. Böhm, W. Hübner, and K. Väänäen, "GIVEN: Gesture Driven Interactions in Virtual Environments: A Toolkit Approach to 3D Interactions," in *Proc. Interfaces to Real and Virtual Worlds*, Montpellier, France, Mar., 1992, pp. 243–254.

[2] W. Chapin and S. Foster, "Virtual Environment Display for a 3D Audio Room Simulation," in *Proceedings of SPIE Stereoscopic Display and Applications*, 1992.

[3] G. Burdea and P. Coiffet, *Virtual Reality Technology*. New York: Wiley, 1994.

[4] K. Shimoga, "Finger Force and Touch Feedback Issues in Dextrous Telemanipulation," in *Proc. NASA-CIRSSE International Conf. Intelligent Robotic Systems for Space Exploration*, Troy, NY, Sept., 1992.

[5] H. Sundgren, F. Winquist, and I. Lundstrom, "Artificial Olfactory System Based on Field Effect Devices," in *Proc. Interfaces to Real and Virtual Worlds*, Montepellier, France, Mar., 1992, pp. 463–472.

[6] J. Mogal, "The SGI Reality Engine," in *Panel on Virtual Reality, CyberArts International*, Pasadena, CA, Oct., 1992.

[7] I. Iwata, "Pen Based Haptic Virtual Environment," in *Proc. IEEE Virtual Reality Annual International Symposium (VRAIS)*, Seattle, WA, Sept., 1993, pp. 287–292.

[8] M. Minsky, M. Ouh-young, O. Steele, F.P. Brooks Jr., and M. Behensky, "Feeling and Seeing: Issues in Force Display," *Computer Graphics*, Vol. 24, No. 2, pp. 235–243, 1990.

[9] VPL Research, "Dataglove Model 2 Operating Manual," Redwood City, CA, 1987.

[10] B. Marcus and P. Churchill, "Sensing Human Hand Motions for Controlling Dextrous Robots," A.D. Little, Boston, 1988.

[11] G. Burdea, J. Zhuang, E. Roskos, D. Silver, and N. Langrana, "A Portable Dextrous Master with Force Feedback," *Presence-Teleoperators and Virtual Environments*, Vol. 1, No. 1, pp. 18–29, 1992.

[12] P. Cutt, "Tactools Technical Summary," Xtensory Inc., Scotts Valley, CA, 1992.

[13] R. Stone, "Advanced Human-System Interfaces for Telerobotics Using Virtual Reality and Telepresence Technologies," in *Proc. 5th International Conf. Advanced Robotics ('91ICAR)*, Pisa, Italy, June, 1991.

[14] S. Jacobsen, E. Iverson, C. Davis, D. Potter, T. McLain, "Design of a Multiple Degree of Freedom, Force Reflective Hand Master/Slave with a High Mobility Wrist," in *Third Topical Meeting on Robotics and Remote Systems*, Charleston, SC, 1989.

[15] N. Patrick, T.B. Sheridan, M.J. Massimino, and B. Marcus, "Design and Testing of a Non-reactive, Fingertip, Tactile Display for Interaction with Remote Environments," *SPIE Cooperative Intelligent Robotics in Space*, Vol. 1387, pp. 215–222, Nov., 1990.

[16] L. Rosenberg, "The Use of Virtual Fixtures as Perceptual Overlays to Enhance Operator Performance in Remote Environments," Technical Report, Center for Design Research, Stanford University, CA, 1992.

[17] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practice*. New York: Addison-Wesley, 1990.

[18] P. Appino, B. Lewis, L. Koved, D. Ling, D. Rabenhorst, and C. Codella, "An Architecture for Virtual Worlds," *Presence-Teleoperators and Virtual Environments*, Vol. 1, No. 1, pp. 1–17, 1992.

[19] G. Burdea, E. Roskos, D. Silver, F. Thibaud, and R. Wolpov, "A Distributed Virtual Environment with Dextrous Force Feedback," in *Proc. Interface to Real and Virtual Worlds Conf.*, Montpellier, France, Mar., 1992, pp. 255–265.

[20] C. Grimsdale, and S. Ghee, "Virtual Realities—Artificial Environments," in *Proc. Computer Graphics Conf. '90*, London, 1990, pp. 1–8.

[21] M. Cutkosky, and R. Howe, "Human Grasp Choice and Robotic Grasp Analysis," *Dextrous Robot Hands*. New York: Springer, 1990, pp. 5–31.

[22] R. Cholewiak, and A. Collins, "Sensory and Physiological Bases of Touch." in *The Psychology of Touch*, M. Miller and W. Schift, Eds. Hillsdale, NJ: Erlbaum, 1991, pp. 23–60.

[23] G. Burdea, J. Zhuang, E. Roskos, D. Silver, and N. Langrana, "Direct-drive Force Feedback Control for the DataGlove," in *Engineering Systems with Intelligence—Concepts, Tools, and Applications*, Spyros G. Tzafestas, Ed. Dordrecht: Kluwer Academic, 1991, pp. 453–460.

[24] G. Burdea and J. Zhuang, "Actuator System for Providing Force Feedback to a Dextrous Master Glove," U.S. Patent 5,143,505, Sept., 1992.

[25] J. Hong and X. Tan, "Teleoperating the Utah/MIT hand with a VPL DataGlove I. DataGlove Calibration," in *Proc. 1988 IEEE International Conf. Robotics and Automation*, 1988, pp. 1752–1757.

[26] E. Roskos and J. Zhuang, "Real-Time Software for the VPL DataGlove," in *CAIP-TR-124*, Center for Computer Aids for Industrial Productivity, Rutgers University, 1990.

[27] Hewlett Packard Co., "Starbase Graphics Techniques: HP-UX Concepts and Tutorials," Vol. 1–3, 1988.

[28] J-P. Gourret, "Simulation of Object and Human Skin Deformations in a Grasping Task," *Computer Graphics*, Vol. 23, No. 3, 1989.

[29] H. Rijpkema and M. Girard, "Computer Animation of Knowledge-Based Human Grasping," *Computer Graphics*, Vol. 25, No. 4, 1991.

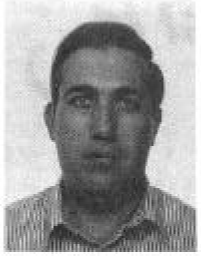
[30] S. Sclaroff and A. Pentland, "Generalized Implicit Functions for Computer Graphics," *Computer Graphics*, Vol. 25, No. 4, 1991.

[31] N. Hogan, "Controlling Impedance at the Man/Machine Interface," in *Proc. 1989 IEEE International Conf. Robotics and Automation*, May, 1989, pp. 1626–1631.

[32] P. Richard, G. Burdea, and P. Coiffet, "Human Performance in Tasks Involving Force Feedback from Virtual Objects," in *Proc. Interface to Real and Virtual Worlds Conf.*, Montpellier, France, Mar., 1993.

[33] P. Richard, G. Burdea, G. Birebent, D. Gomez, N. Langrana, and P. Coiffet, "Effect of Frame Rate and Force Feedback on Virtual Object Manipulation," in *Presence—*

Teleoperations and Virtual Environments. Cambridge, MA: MIT Press, 1995 (accepted).



Grigore Burdea is Associate Professor in the Department of Electrical and Computer Engineering, and Director of the Human–Machine Interface Laboratory of the CAIP Center at Rutgers University. His research focuses on force/tactile feedback for virtual reality, where he published the book, "Virtual Reality Technology" (John Wiley & Sons). Dr. Burdea is Co-Editor of "Computer Integrated Surgery" (to be published by MIT Press) and Associate Editor of the *IEEE Transactions of Robotics and Automation*.

Daniel Gomez is currently a PhD student at the Human–Machine Interface Laboratory at Rutgers University. He received his BS in Physics from Universidad Nacional de Colombia (Bogota, Columbia) and his MS in Electrical Engineering from Rutgers University. His research interests include virtual reality and robotics. He is currently working on developing a hand master with force feedback for virtual reality simulations.

Noshir Langrana is Professor in the Department of Mechanical and Aerospace Engineering, Director of the Computer Aided Design Laboratory, and a Member of the Biomedical Engineering Graduate Program at Rutgers University. His research focuses on force feedback in virtual reality. Dr. Langrana is a Fellow of the American Society of Mechanical Engineers, a Registered Professional Engineer (PE), and a Member of the Orthopaedic Research Society.

Edward Roskos received a BA in computer science and a BS in Electrical Engineering in December 1991 and is currently finishing the Masters Program in Computer Science at Rutgers University. Mr. Roskos is currently working with AT&T in Holmdel, New Jersey and was a staff member of CAIP (Computer Aids for Industrial Productivity) where he designed and implemented virtual reality software. His main interests include discrete mathematics, software development, virtual reality, and distributed processing.

Paul Richard is currently a PhD student at the Laboratoire de Robotique de Paris. He received his BS and MS in Electronics and Automation from University of Nice, Sophia-Antipolis in France. His research activities and interests are robotics, virtual reality, and human factors. He is working on human factors involved in virtual environment, namely visual perception and haptic feedback. He has investigated the influence of haptic sensory substitution and information redundancy on human performance in virtual environment. He also worked at the Human–Machine Interface Lab at Rutgers University as a visiting researcher.