A DISTRIBUTED VIRTUAL ENVIRONMENT WITH DEXTROUS FORCE FEEDBACK

Grigore Burdea, Edward Roskos, Deborah Silver,
François Thibaud & Robert Wolpov

Rutgers University
Piscataway, New Jersey, USA

Abstract *: Force feedback is a welcome enhancement of dextrous masters used in virtual environments, telerobotics, entertainment, or medicine. This paper describes work towards a distributed system built around the Rutgers Portable Force Feedback Master. This master gives force feedback to multiple fingers of the Dataglove (VPL Research Inc). The system also uses two Sun 4 workstations and a HP graphics workstation communicating over the ethernet. Several examples are presented for virtual dextrous force feedback.*

Keywords *: force feedback, portable master, Dataglove, distributed system, object model, object compliance.*

1. Introduction

Commercially available dextrous masters such as the Dataglove [1] and the Exos Dextrous Hand Master [2] are currently used in virtual environments simulation, telerobotics, medicine and entertainment. One major drawback is that these masters operate in open-loop, with no force/touch feedback to the operator.

Work to bring force feedback to dextrous masters has resulted in a number of prototypes such as the Teletact Glove [3] and the Portable Force Feedback Dextrous Master [4]. Both systems are compact, light, desk-top and differ greatly from more complex (and more expensive) systems developed exclusively for telerobotics [5].

Work on the hardware aspects of force feedback for virtual environments is well underway. However, research on communications, control and graphics simulation with force feedback has just begun. Several research groups in USA and Europe are working on Virtual Environments Distributed Systems [6,7,8,9,10]. However none have to date fully integrated force feedback to virtual object grasping.

This paper presents work towards a distributed system for virtual environments that integrates force feedback. As such it goes beyond any of the systems previously cited. Section 2 describes the experimental set-up. Section 3 presents the virtual environment distributed software. Section 4 models virtual force feedback generated when grasping objects such as springs, balls and soda cans.

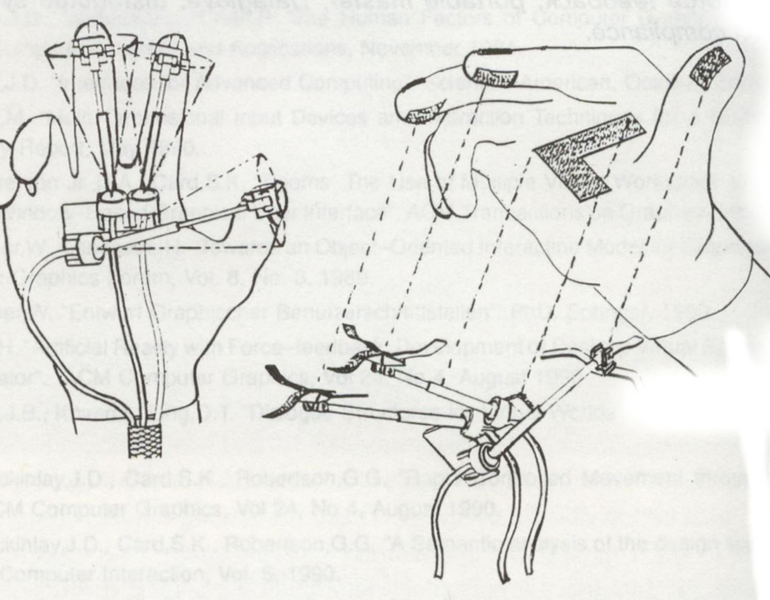

Figure 1: The Portable Force Feedback Dextrous Master [4]

2. Experimental system set-up

The Rutgers Portable Force Feedback Dextrous Master is illustrated in Figure 1. The compact feedback structure has three actuators placed in the palm of a Dataglove. This "direct-drive" system enjoys the same advantages as those used to drive robot arms, namely the absence of cables or pulleys and reduced friction. In order to allow the abduction-adduction of the fingers (thumb, middle and index), three sphere joints were placed on a small L-shaped platform attached to the exterior of the glove. Three cylinders were mounted coaxially with these sphere joints, thus allowing for direct connection to air tubes passing through the sphere joints.

Each cylinder shaft was attached to the glove finger tips through cylindrical joints that allow movement in a plane normal to the fingertips. The attachement of the feedback structure to the glove was done with velcro™ strips mounted in the palm and on the fingers. These detachable connections allowed for adjustments to the hand characteristics of different users.

The actuator controllers and air valves were grouped in a modular hardware interface unit. This interface consists of three proportional pressure regulators, a pressure indicator for the main air input, power supply (24 V) and D/A input port. The interface dimensions are modular with the Dataglove electronic unit so the two units can be "stacked-up" to save desk space.

The configuration of the experimental system used to integrate the master in a virtual environment is shown in Figure 2. The feedback actuator interface was connected to a D/A

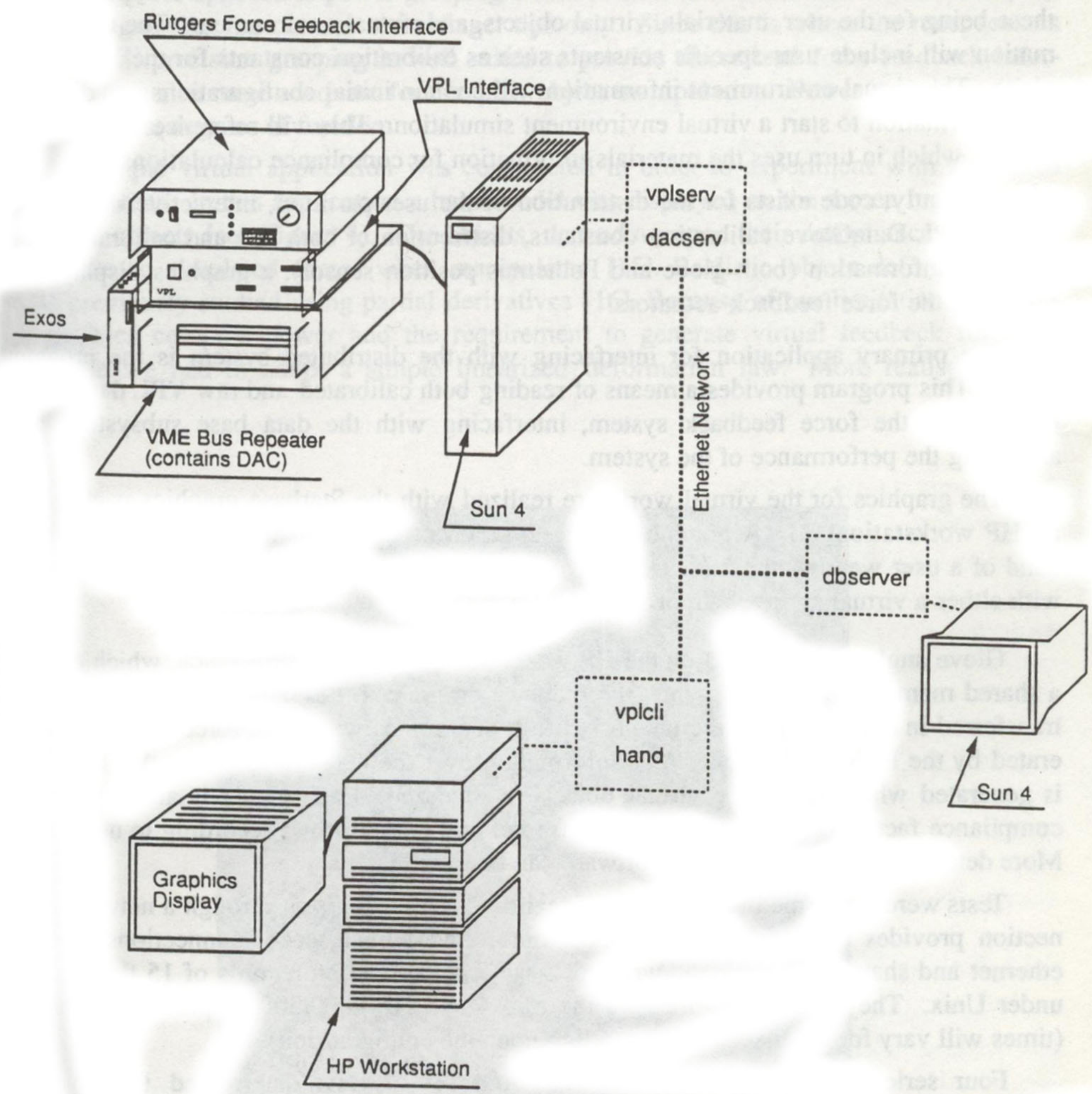

Figure 2. Experimental system set-up [4]

board of a Sun 4 host. The Dataglove VPL interface box was connected to the same Sun 4 through a serial port. A separate HP 9000 series 360 workstation was used for 3-D color graphics. A second Sun 4 workstation was used as a database server for the other two computers. Intercommunication between the three computers was done over the ethernet and utilized a master/slave process distribution.

3. The Virtual Environment Software

A distributed, object oriented software collection is being developed to simulate the virtual environment and control the input and output devices[11]. The need for distributing the load stems from the amount of real-time computation required. Tasks to be done include reading input devices (dextrous master in our case), updating the velocity of objects in motion, handling collision detection between objects, determining hand orientation, computing object deformations and resulting contact forces, and updating the actuator pressures.

Our data base is distributed from a localized server and consists of four types of data, these being for the user, materials, virtual objects, and virtual environment. The user information will include user-specific constants such as calibration constants for the VPL Dataglove. The virtual environment information will contain initial configurations and distribution information to start a virtual environment simulation. This will reference object information, which in turn uses the materials information for compliance calculations.

Currently, code exists for the distribution of the user partition, interactive determination of VPL DataGlove calibration constants, distribution of both raw and calibrated VPL interface information (both glove and Polhemus position sensor), a graphics display, and update of the force feedback actuators.

The primary application for interfacing with the distributed system is the program rcalib. This program provides a means of reading both calibrated and raw VPL data, interfacing with the force feedback system, interfacing with the data base subsystem, and analyzing the performance of the system.

The graphics for the virtual world are realized with the Starbase graphics package on an HP workstation[12]. A graphical representation of a human hand corresponds to the hand of a user wearing the VPL Dataglove with Polhemus sensor. The user may interact with either a virtual spring, ball, or soda can.

Glove angles are received on the HP workstation by the program vplcli, which creates a shared memory segment to which the graphics program, hand, attaches. Using the data transferred in real-time over the network, hand determines whether a force should be generated by the PDMFF and relays this information over the network to dacserv. The force is generated when one of the virtual objects is squeezed. Each object is attributed with compliance factors and graphically deforms (and possibly reforms) according to user input. More details regarding the system software can be found in [13].

Tests were performed to determine if real-time data being sent through a network connection provides sufficient bandwidth. Sample server/client socket connections over an ethernet and shared memory read/write tests were performed on records of 15 float values under Unix. They were performed on the Sun 4 and HP 9000/300 graphics workstation (times will vary for other systems depending upon the configuration).

Four series of tests were made. The first set of tests determined the possible bandwidth of reading records from a data file. The second set of tests read from a file and stored the values in a shared memory segment. Third, the Sun 4 server read from a file and sent the values over the ethernet to a client. Lastly, a server read from a file, sent the values over the ethernet, and the client wrote the results into a shared memory segment. The results are summarized in Table 3. Tests showed that the network allows sufficient bandwidth for process distribution. The two bottlenecks remaining are the Dataglove sampling rate of about 30 Hz and the HP graphics refresh rate.

	Sun 4	HP
File	371 Hz	138 Hz
File & ShM	365 Hz	136 Hz
File & Net	191 Hz	155 Hz
File, Net, & ShM	178 Hz	152 Hz

Figure 3: Summary of performance for sending dummy VPL records

4. Virtual Object Modeling with Force Feedback

The task modeled by our computer simulation involves precision grasping of light virtual objects. As opposed to power grasping where contact is with the whole palm area [14], precision grasping has contact at the finger tips only. Since this is where the most sensors are located, precision grasping is well suited to provide discretized force feedback information. It is also more adequate for complex object manipulation. Power grasping utilizes only the wrist degrees of freedom.

A simple virtual application was constructed in order to experiment with simulated contact forces. In this application a hand model was displayed on the screen together with modeled objects (see Figure 4). The objects, depending upon their compliance, deformed and returned simulated forces when manipulated [15]. Realistic object deformation has been previously studied using partial derivatives [16]. Because of our limitations in terms of graphics compute power and the requirement to generate virtual feedback forces in real-time we had to adopt a simple linearized deformation law. More realistic object

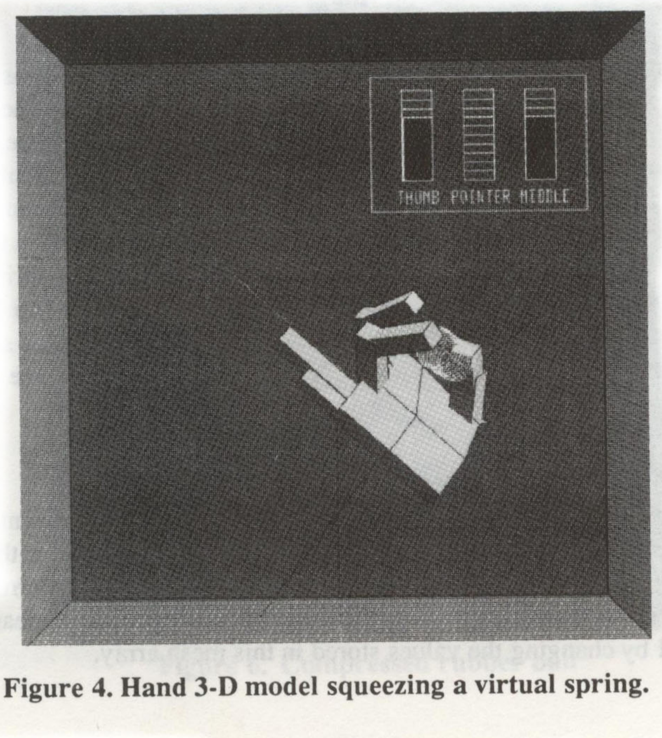

Figure 4. Hand 3-D model squeezing a virtual spring.

deformation can be accomplished using higher orders (e.g. finite element methods [16], [17]). In our system the simpler graphics feedback is compensated by the force feedback loop, for improved man-machine interfacing. In the next Section we describe these sample objects and some of the force issues which arise.

The hand model has been simplified as well (see for example [16]) in order to improve the graphics refresh rates. The distal and middle phalanges were combined for both the index and middle fingers. Thus, the number of joints was reduced from three to two for both fingers. The finger which had the fewest degrees of freedom was the middle finger. It had two joints with one flexion angle each. The index finger had the same flexion angles as the middle, in addition to an abduction/adduction angle for the Metacarpophalangeal joint. The thumb had the degrees of freedom of the index finger plus the anteposition/retroposition angle. The Polhemus sensor data were used to provide three palm translations and three rotations.

During each display cycle all the Dataglove position data are transmitted through the network and the shared memory to the HP workstation. The graphics package keeps fetching data and rendering image on the screen. Thus, within each display cycle both the hand gesture and the shape of the object are updated. This leads to the continuous contact force changes according to the object deformation.

The contact force feedback to the human operator can be scaled either up and down to satisfy different touch sensation to the finger tips for different stiffnesses of the manipulated objects. The scaling of force can be implemented in two ways. One way is to manually adjust the internal settings of the current regulators inside the interface box. These settings can reduce or expand the range of electric current input to the proportional valves. So the maximum force can be adjusted in the desired range. Another way is to characterize the different objects within the software package. In order to view the magnitudes of contact forces on the three finger tips directly, a force indicator is created on the graphics screen which contains three bars presenting the force scale for each individual finger tip.

4.1. Sample objects: Spring, Ball, Soda Can

4.1.1 Spring

The simplest model used to test the force reaction under virtual grasping is a generic spring with two ends squeezed between the thumb and middle finger. The tip of the thumb and middle finger define a pinching line [18]. During each display cycle, the positions of both finger tips are determined and the distance in-between them calculated. If the distance is greater than the spring undeformed length, no contact forces occur. Otherwise, the contact force is generated according to Hooke's law $F = K\,\Delta x$. Here F is the contact force, K is the specified spring stiffness and Δx is the difference between the spring length and the distance between the thumb and the middle finger tips. The relationship between the contact force and the deformation is characterized in Figure 5-a. As can be seen the feedback force is present even during grasp relaxation, so the user will feel a force when opening or closing the hand.

4.1.2. Rubber Ball

The requirement for small refresh time led to the rubber ball representation by a single quadrilateral mesh of radius R. The quadrilateral mesh is stored as a three dimensional array, *mesh[row][column][index]* where *index* ranges from 0 to 5 corresponding to the x,y,z data of the mesh point and its vertex normal. Changes in the appearance of the ball are accomplished by changing the values stored in this mesh array.

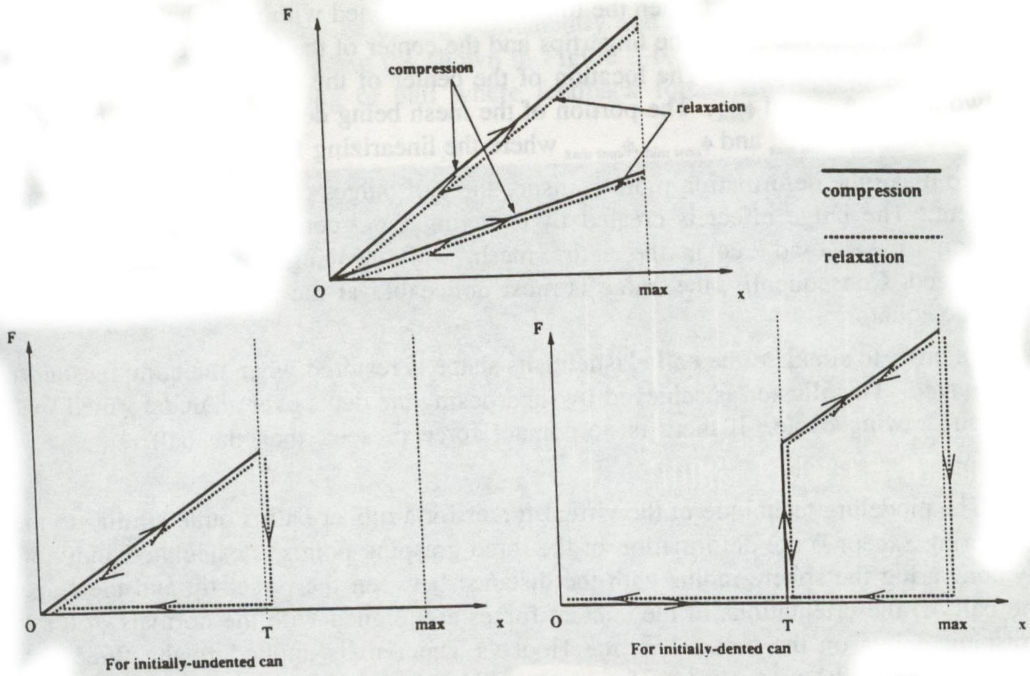

Figure 5. Contact force vs. virtual object deformation

When the ball is deformed, selected vertices will be remapped to the surface of a smaller sphere. This can be accomplished by multiplying each vertex component within the compressed region by a depth factor, $0 \leq depth \leq 1$. The value of this depth parameter is equal to the distance between the fingertip and the center of the ball, divided by the sphere's radius.

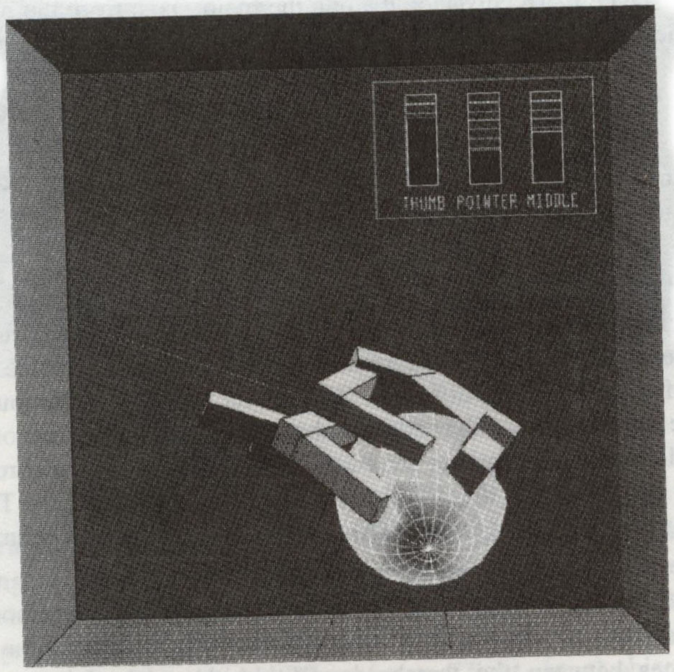

Figure 6. Compressed rubber ball

The ball is compressed when the fingertip is positioned within a sphere radius distance from the center of the ball. The fingertips and the center of the ball are each represented as a three-dimensional point. The location of the center of the compressed area is specified by two angles θ_{com} and ϕ_{com}. The portion of the mesh being deformed is given by the range of angles $\theta_{com\,min}, \theta_{com\,max}$ and $\phi_{com\,min}, \phi_{com\,max}$ where the linearizing factor is $0.6 \times arccos(depth)$.

To make the deformation more realistic, the ball bulges proportionally to the depth of the dent. The bulge effect is created by changing the z-component of each vertex as a function of $depth$ and $\cos\theta$ in the entire mesh, while leaving the x and y components unchanged. Consequently, the bulge is most noticeable at the poles and least noticeable near the equator.

In order to simulate the ball elasticity its shape is restored when the compression force is removed. This illusion is achieved by superposing the dents over an undeformed sphere at each drawing cycle. If there is no contact force present, then the ball remains undeformed.

The modeling technique of the virtual forces for a rubber ball is quite similar to that of the spring except i) the deformation of the three grasping points is calculated individually by comparing the sphere radius with the distance between the finger tip and the center of the ball. ii) the orientations of the contact forces are aligned with the normals of the three grasping points on the ball. Thus, the Hooke's Theorem is applied to the three contact forces individually, as $F_i = K\,\Delta x_i$, for $i = 1, 2, 3$. Here i denotes the three contact forces for the thumb, the index and middle fingers. The stiffness is approximated constant throughout the ball.

4.1.3. Soda Can

The vertices for the static meshes were calculated using sketches of the desired shapes and trigonometric relations. The points along the mesh are calculated using the standard cylindrical coordinate formulas. The soda can becomes dented when the fingertip is within one radius distance from the can's major axis. The location of the dent is specified by an angle ϕ and a distance h. The distance h along the major axis, from the bottom of the can, is $h = \sqrt{l^2 - d^2}$, where l is the distance from the contact point to the can bottom and d is the cylinder radius. The angle ϕ is found using the dot product of the x-direction unit vector and the vector which extends from the major axis to the surface in contact with the fingertip.

The soda can denting routine works by flattening portions of latitudinal slices of the mesh from ϕ_{min} to ϕ_{max}.

4.2. Discussion

We have chosen these three objects because of their simplicity, diversity and ease of human-factor tests. The graphics requirement may look similar for the ball and soda can, but their force feedback is very different. The rubber ball has continuous compliance, and has only elastic deformations. The soda can, however, has two regions of deformation. For relatively small dents, deformations are elastic, and the local region is restored to its original shape. For deeper dents, the deformation is plastic (permanent). This requires that a list of dent information be maintained. The permanent dents are not returned to the original shape.

The range of deformations is subdivided into a number of thresholds. The grasping hand will feel the contact forces only when dynamically squeezing the can or within the elastic sub-ranges between the thresholds. Within these elasticity ranges the Hooke's theorem is applied as previously discussed. Once the fingers stop squeezing beyond a plastic threshold, the can will then be dented permanently and feedback forces are not present during the relaxation of the grasp (as shown in Figure 5-b,c). This is a different response than that of the virtual spring or ball where feedback forces are present during grasp release.

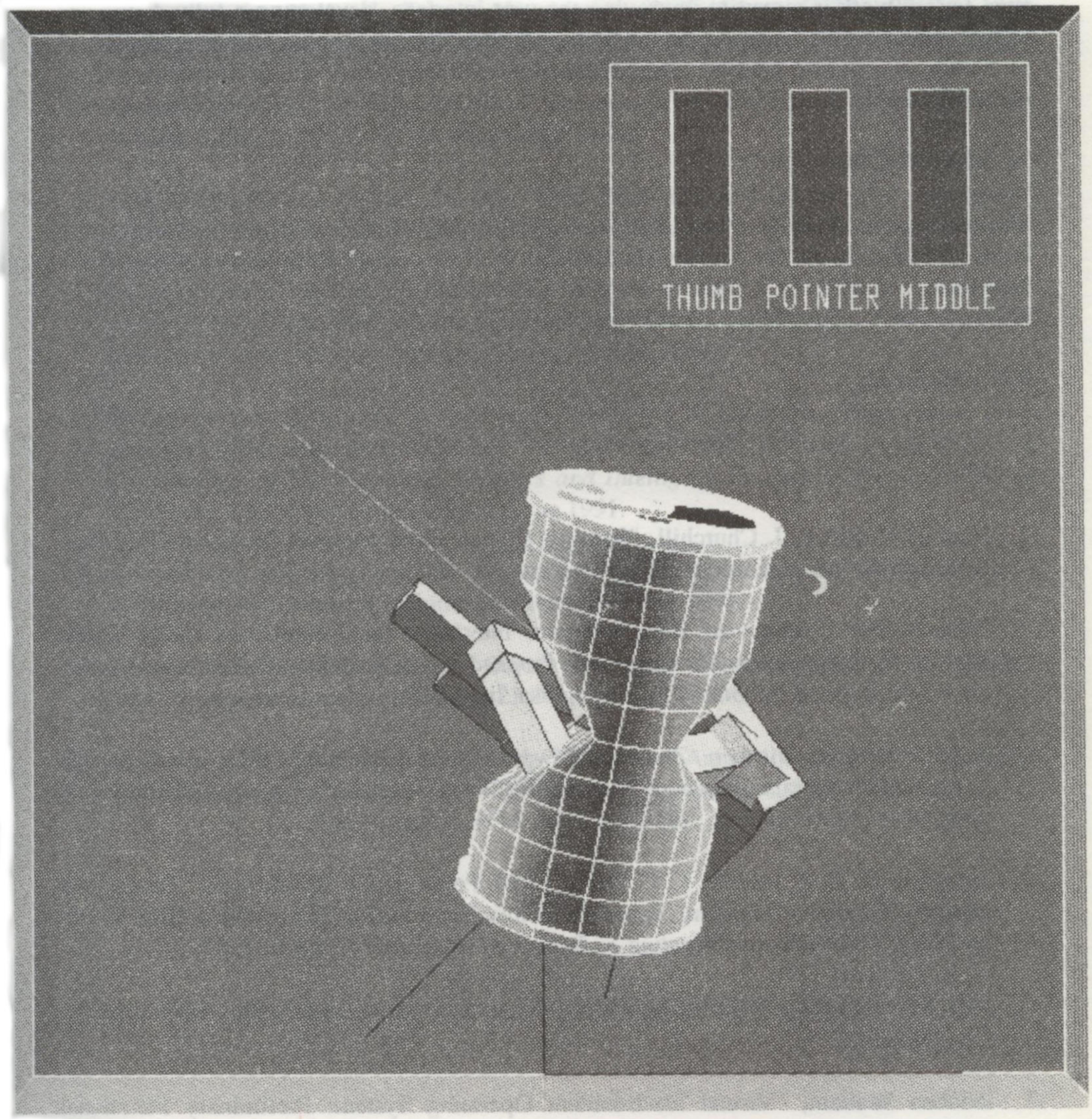

Figure 7. Dented soda can

5. Conclusion

Precision grasping is the manipulation of objects using multiple fingertip contacts. Precision grasping of virtual objects is enhanced by the addition of force feedback to the user hand. A system using a Portable Force Feedback Dextrous Master, two Sun 4 workstation and a HP graphics workstation was presented. The processes are distributed over the network in order to improve bandwidth. Three types of light virtual objects were modeled.

Two have elastic deformation over the full range of motion (spring and ball) while the third has both elastic and plastic deformation. This changes the laws generating virtual feedback forces. Forces displayed in the virtual model as bar graphs determine the pressure in the master feedback actuators. Work is still in the initial stages and the graphics is simplified, however we feel that force feedback is a definite plus for the realistic simulation. Future work is aimed at improving the user interface, developing a fully distributed virtual environment system, the addition of other virtual objects, more sophisticated modeling, and human studies. The human studies will quantify the performance change due to virtual force feedback.

ACKNOWLEDGMENTS

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 a Special Purpose Grant from the A.T.&T. Foundation, and by National Science Foundation Grants MSS-89-09335 and CCR-89-09197.

REFERENCES

1. ''Dataglove Model 2 Operating Manual,'' VPL Research, Inc., Redwood City, CA. 1987.

2. Marcus B. and P. J. Churchill, "Sensing Human Hand Motions for Controlling Dextrous Robots", A. D. Little, 1988.

3 Stone Robert, "Advanced Human-System Interfaces for Telerobotics Using Virtual Reality & Telepresence Technologies", *Proceedings of the Fifth International Conference on Advanced Robotics ('91ICAR)*, Pisa, Italy, 1991.

4. Burdea Grigore, Jiachen Zhuang, Edward Roskos, Deborah Silver, and Noshir Langrana, "A Portable Dextrous Master with Force Feedback", *Presence-Teleoperators and Virtual Environments*, MIT Press, Vol. 1 (1), January 1992.

5 Jacobsen, S., E. K. Iversen, C. C. Davis, D. M. Potter and T. W. McLain, ''Design of a Multiple Degree of Freedom, Force Reflective Hand Master/Slave with a High Mobility Wrist,'' *Third Topical Meeting on Robotics and Remote Systems*, 1989.

6 VPL Research Inc., "RB2 Virtual Reality System", Product Description, Redwood City, CA, July 1989.

7 Bricken William, "Virtual Environment Operating System: Preliminary Functional Architecture", Human Interface Technology Laboratory, TR-HITL-M-90-2, University of Washington, Seattle, 1990.

8 Moshell Michael, Charles Hughes, Brian Blau, Xin Li and Richard Dunn-Roberts, "Networked Virtual Environments for Simulation and Training", *Proceedings SimTec 1991 Conference*, Society for Computer Simulation, Oct. 1991.

9 Grimsdale Charles and Stephen Ghee, "Virtual Realities - Artificial Environments", *Proceedings of the Computer Graphics Conference '90*, London, UK, pp1-8, 1990.

10 Appino Perry, Bryan Lewis, Lawrence Koved, Daniel Ling, David Rabenhorst and Christopher Codella, "An Architecture for Virtual Worlds", *Presence-Teleoperators and Virtual Environments*, MIT press, in print.

11 Roskos Edward, "Towards a Distributed, Object-Oriented Virtual Environment", Report to the National Science Foundation, NSF Grant CCR89-09197, 18pp, November 1991.

12 Hewlett Packard, "Starbase Graphics Techniques HP-UX Concepts and Tutorials" Volume 1, Hewlett-Packard Co., December 1988.

13 Roskos Edward and Jiachen Zhuang, "Real-Time Software for the VPL Dataglove", CAIP-TR-124, Center for Computer Aids for Industrial Productivity, Rutgers University, September 1990, 73pp.

14 Cutkosky Mark and Robert Howe, "Human Grasp Choice and Robotic Grasp Analysis", in *Dextrous Robot Hands*, Springer-Verlag, pp 5-31, 1990.

15 Burdea Grigore, J. Zhuang, E. Roskos, D. Silver and N. Langrana, "Direct-drive Force Feedback Control for the Dataglove", *Proceedings of The European Robotics and Intelligent Systems Conference*, Kluwer Academic Publishers, Corfu, Greece, June 1991.

16 Sclaroff Stan and Alex Pentland, "Generalized Implicit Functions for Computer Graphics", *Computer Graphics*, Volume 25, Number 4, July 1991.

17 Gourret Jean-Paul, "Simulation of Object and Human Skin Deformations in a Grasping Task", *Computer Graphics*, Volume 23, Number 3, July 1989.

18 Rijpkema Hans and Michael Girard, "Computer Animation of Knowledge-Based Human Grasping", *Computer Graphics*, Volume 25, Number 4, July 1991.