Virtual Reality Simulation Modeling for a Haptic Glove

V. Popescu, G. Burdea, and M. Bouzit*
Department of Electrical and Computer Engineering
*Center for Computer Aids for Industrial Productivity (CAIP)
Rutgers – The State University of New Jersey,
Piscataway, NJ 08854, USA.
http://www.caip.rutgers.edu/vrlab
Phone: (732) 445-0561/4775 (fax)

Abstract

The recent addition of force and touch feedback to Virtual Reality simulations has enhanced their realism. Research on haptics interfaces is now extended to physical modeling of contact surfaces, object hardness, surface deformation, etc.. This is especially needed when dextrous manipulation of virtual objects is concerned. This paper describes a VR system using a haptic glove (Rutgers Master II) connected to a PC workstation, and a new method for modeling virtual hand haptic interactions. An application example presented here is an orthopedic rehabilitation library. The exercises in this library involve interactions with dynamic objects and physical modeling of plasticity.

Keywords: Virtual Reality simulation, haptic rendering, haptic interaction mesh, Rutgers Master II glove.

1. Introduction

Haptic (force and touch) feedback is a recent addition to Virtual Reality simulations [3]. This sensorial modality increases the simulation realism during virtual object manipulation. Haptic feedback is mandatory when the graphics is corrupted (simulating poor visibility) or when the manipulated object is partly or totally occluded, or when the environment is dark.

The addition of haptics requires several enhancements to the typical VR simulation. First the system needs a dedicated haptic interface which transmits the feedback to some part of the user's body (usually the hand). Most of today's interfaces are joysticks, or small robotic arms. Second, physical modeling tools need to be developed, containing a library of functions to calculate and replicate contact forces, surface deformation, rugosity, object weight, etc.

Haptic rendering tools were usually designed for specific interface devices. One of the most successful haptic interfaces commercially available is PHANToM [7]. This small robotic arm provides forces (but no torques) to the user's index finger. Therefore most of the papers modeling the PHANToM are concerned with only the end effector point of contact (known as haptic interface point or HIP) [15] [6]. The ray-based haptic rendering technique proposed by Basdogan et al. take into account haptic instrument shape [1]. The method approximates the pen shape of haptic interface end effector with a line segment. The pen shape is taken into consideration for collision detection checking and force calculation. Additionally, a force shading method for smoothing force interactions with polygonal shapes is described. State of the art of haptic interaction modeling is presented in [12].

A more complex category of haptic interfaces is haptic gloves. Several haptic gloves were designed for VR simulations: the "Rutgers Master" [4], the LRP Glove [2], the "CyberGrasp" [13], etc.. Force feedback bandwidth for these devices is in the range of 10 - 50 Hz. These gloves have one or more force degrees of freedom (DOF) per finger with forces grounded in the palm or on the back of the hand. A virtual hand maps the user's hand to the Virtual Environment.

Hand interactions with virtual objects require a more elaborate model (than the HIP) due to the complex shape of the virtual hand. This paper proposes a hand interaction model that accounts for the shape of interacting virtual objects. The proposed force mapping and object deformation is based on the *haptic interaction mesh* model. Section 2 presents the VR simulation experimental setup and haptic interface control loop. Section 3 describes the haptic interaction model and haptic rendering methods for VR simulations. Section 4 illustrates our approach with simulations of an orthopedic rehabilitation application.

Section 5 concludes the paper and gives future work directions.

2. The VR simulation system

The main components of our VR simulation are the graphics and the haptics systems. The graphics runs on a PC workstation (Powerdigm XSU – 300 MHz) with a FireGL 4000 graphic accelerator (rendering up to 2M triangles/sec). The haptic system is composed of Rutgers Master II (RM-II) glove connected to a Haptic Control Interface (HCI). The RM-II glove is a portable haptic interface designed for dextrous interactions with virtual environments [4]. The interface uses an embedded Pentium PC (currently 100 MHz) to control forces applied at user’s fingertip, read sensor data, and communicate with the host PC. The software control loop running at 500 Hz assures a precise and stable output force at user fingertip [8]. Additionally the software running on the embedded Pentium reads and filters data from the RM-II sensors through an A/D board and transforms it into user's hand joint angles. Communication between the HCI and the host computer is through an RS232 port. Data sent to the host computer contains joint angles, measured forces or device state. Received data includes commands or forces to be displayed to the user. At a rate of 38,400 bits/second the RS232 line can transmit up to 136 RM-II position data sets or 120 position and force data sets every second. The software control modules are presented in Figure 1.

The haptic rendering engine was implemented using two different models:

a) local haptic rendering – the forces are all calculated and displayed locally on the HCI, based on some parameters downloaded from the host computer;

b) distributed haptic rendering – the forces are calculated on the host and transmitted to the HCI to be displayed to the user fingers.

In the first model the PC workstation only commands the beginning and the end of haptic feedback loop. The object database contains the models of hapticly active objects (ball, power putty, etc.). Haptic interaction parameters (spring constant, friction and dumping constants, position parameters) are transmitted by the host PC when the force feedback loop is activated. Forces are subsequently calculated locally on the HCI based on sensor readings from the RM-II glove (finger position) and object deformation models. In the second case the haptic rendering model resides entirely on the PC workstation. Here the host PC calculates and sends force targets to be displayed by the HCI to user fingers.

The limitation of the first method resides in the number and complexity of models that can be stored in the object database. It is only suited for grasp-release type of interactions as it assumes that the relative position of the hand and grasped object does not change. The limitations of the second method are related to the communication bandwidth that can be achieved over a serial line. In our experiments the maximum refresh rate of forces to be displayed at the fingertip was only 100 Hz (vs. 500 Hz achieved by the local rendering loop).

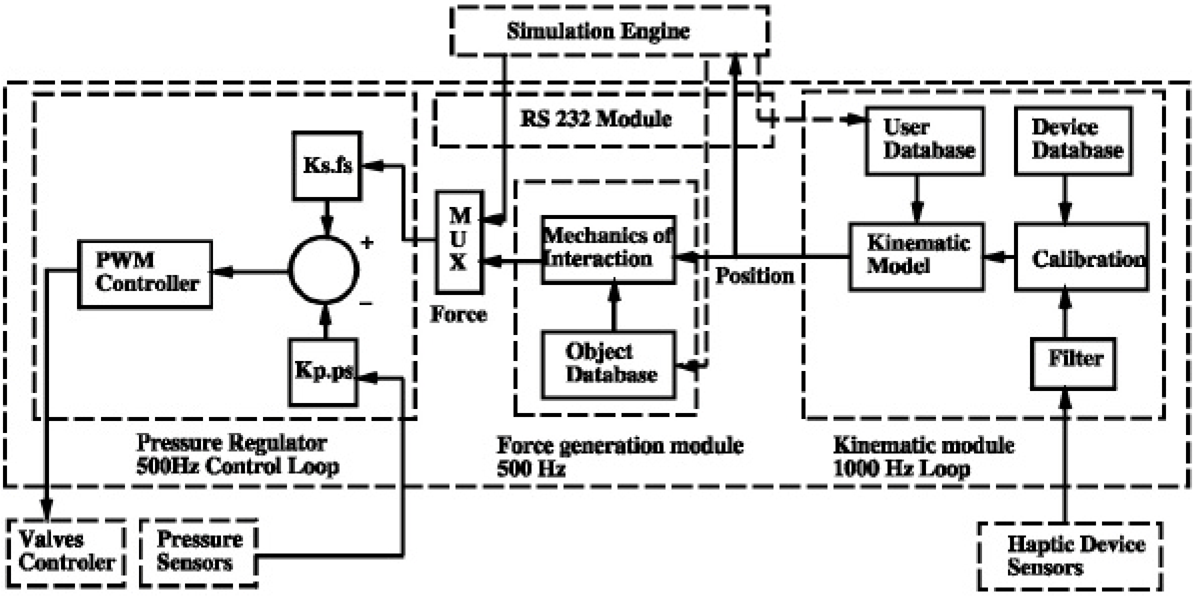


Figure 1: Real-time software of the Haptic Control Interface

3. Hand haptic interaction models

3.1. Virtual hand interaction

Current haptic rendering methods use a HIP for haptic interactions modeling. These methods do not take into account the shape of the haptic instrument. Another method used for a pen shape haptic instrument approximates the end effector with a line and employs a ray based haptic rendering technique. However, in many cases the haptic instrument has a more complex shape that needs a better representation for haptic rendering purposes. An example is a virtual hand for real-time interaction with the VE. Fingers of a virtual hand (which map the user's fingers) have a complex shape that cannot be reduced to simple point or line geometrical primitives.

In what follows we generalize the haptic interface point technique to account for the haptic instrument shape. We propose the use of a *haptic interaction mesh* – a set of points used for haptic rendering and deformation. The *haptic interaction mesh* can be seen as a simplified geometrical shape that captures the essential features of the haptic interface interactions. This has also the role of decoupling the graphic and haptic representations of the haptic interface instrument. The decoupling is beneficial in distributed simulations where the haptics and graphics engines run on different machines. As a result, the level of detail of graphics and haptic models (mesh) can be varied independently according to simulation parameters (haptic interface resolution, real-time simulation constraints, etc.). Generally, the graphics model has a higher resolution due to better graphics display tools available nowadays.

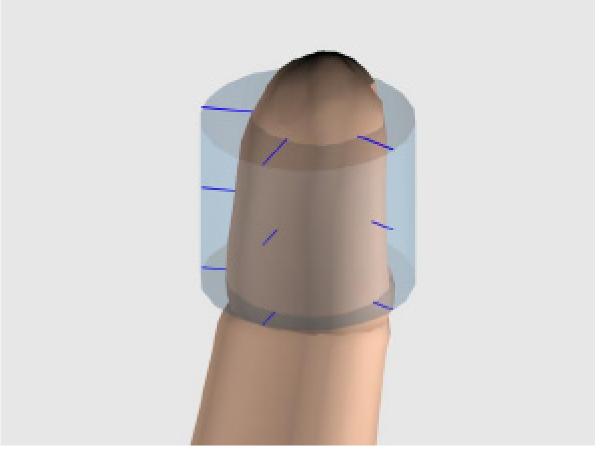


Figure 2: Haptic interaction mesh mapped on a virtual fingertip

The *haptic interaction mesh* representation was applied for modeling the haptic interactions of a force feedback glove. We assumed that the haptic interaction takes place at the fingertips of the virtual hand. Haptic meshes were therefore mapped to each fingertip using the following procedure: a cylindrical mesh with the size of the fingertip is generated; the center of the *haptic interaction mesh* corresponds to the projection of the fingertip center of gravity along a normal pointing towards the palm; a mesh of fixed dimension is selected around this center; the mesh points are then projected on the surface of the fingertip, as illustrated in Figure 2. The generation of the *haptic interaction mesh* is based on the cylinder like shape of the fingertips. The automatic generation of the *haptic interaction mesh* starting from the graphic model of the haptic interface instrument requires further investigation.

3.2. Haptic rendering

Haptic rendering is implemented in three steps: force calculation, force smoothing (shading) and force mapping. As in the HIP method, each haptic point of the *haptic interaction mesh* has a corresponding "ideal haptic interface point" (IHIP), or surface point, on the original (not deformed) surface. The HIP haptic rendering is implemented following the "point-based haptic interaction algorithm" described in [6]. Force calculation uses Hooke's deformation law and the HIP-IHIP displacement vector. Force vectors are first calculated in each point of the *haptic interaction mesh* - *HM_m* - with the formula:

$$\vec{F}_{HM\_m} = k * d_m * \vec{N}_{surface\_m}$$

$$d_m = u\left(< \vec{P}_{surface\,point\_m} - \vec{P}_{HM\_m}, \vec{N}_{surface\_m} >\right)$$

where k represents the object stiffness, d_m is the distance between the surface point and the haptic point along the normal defined at the surface point, and u is the unity step function (forces are applied only when the surface is being deformed).

It is obvious that representing the fingertip with only one HIP is inadequate for force calculation purposes. For example, this representation will not account for the orientation of the virtual fingertip: for the same penetration distance but different fingertip orientations, the same force is displayed at the user's fingertip. The *haptic interaction mesh* method will account for different fingertip orientation. The method is able to provide a better representation of finger haptic interactions provided that the surface of the interacting virtual object has an adequate level of detail. Quantitative measurements of haptic display improvement by using *haptic interaction mesh* method are not available at this time.

Force smoothing (or force "shading") was also implemented in order to avoid large variation in the force displayed when interaction points are crossing edges of polygonal surfaces. The algorithm is similar to Phong shading in graphics [5]. To obtain the normal vector ($N_{surface\_m}$) a weighted average of the vertex normals closest to the point of contact is calculated. The weights are set based on the distance from the point of contact (the closest vertex receives the largest weight). The weighted average normal is then normalized. The normal vector obtained from this calculation is used to determine the direction of the force the surface exerts on the finger. This normal is then used as the projection vector for the calculation of the finger penetration distance.

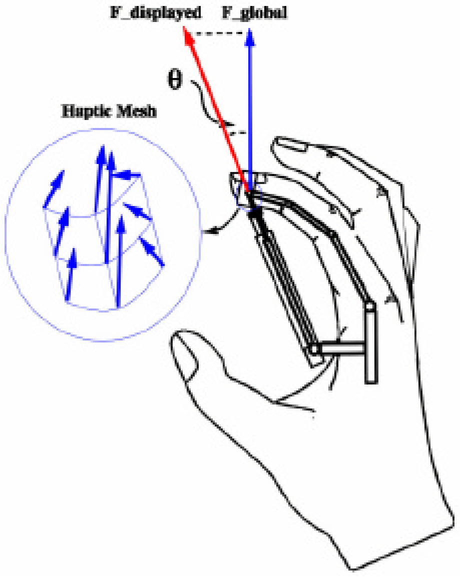


Figure 3: Force mapping for the RM-II glove

The force vectors computed using this method is then mapped to the force displayed to the user's fingertip, as illustrated in Figure 3. In our case, we compute a global interaction force, which represents the projection of the force displayed to the fingertip:

$$\vec{F}_{displayed} = \left(\sum_{m} \vec{F}_{HM\_m} \right) \Big/ \cos q$$

3.3. Surface deformation

Our VR simulation assumes that the virtual finger is rigid, while the other objects in the VE can be deformed. The deformation model implemented uses the displacement vector of the mesh points (see [3] for a review of physical deformation methods). The method is simple, can be executed in real-time and fits well with the haptic rendering techniques previously described.

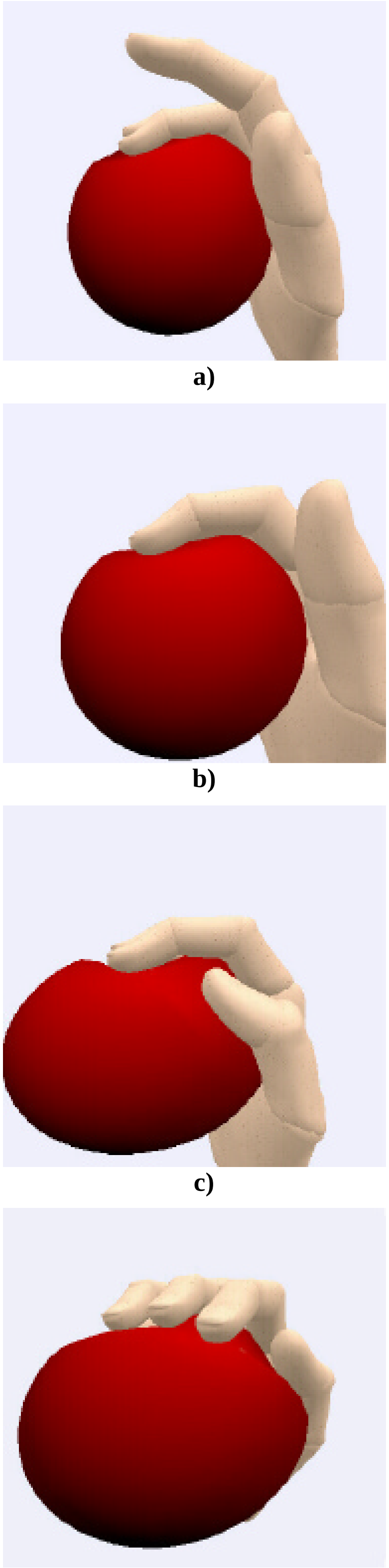

a)
b)
c)
d)

Figure 4 a)-d) Rubber ball squeezing sequence.

The model can be applied to both elastic and plastic deformations. The elastic deformation model uses a non-deformed reference object, while the plastic deformation model updates the reference object after each simulation frame. The elastic deformation is implemented in two steps: a global and a local deformation. The global deformation during grasping and squeezing of virtual objects is implemented using a morphing technique. The vertex position, normals and color are interpolated linearly between a normal state (corresponding to open hand) and a maximum deformation state (corresponding to fully close hand). The interpolation parameter is calculated as a function of finger joint angles:

$$a = \sum_{i=0}^{joints\,\#} q_i \Bigg/ \max\left(\sum_{i=0}^{joints\,\#} q_i \right)$$

The local deformation model is controlled by the fingertip *haptic interaction mesh*. Contact distances are calculated between the points of the *haptic interaction mesh* on the surface of the fingertip and the closest vertices of the intersecting object within a certain radius of influence (typically of the size of the largest dimension of the fingertip bounding box). Deformations are calculated as a function of the distance from the mesh points to the vertices of the deformed object. A second-degree polynomial is currently used to compute the magnitude of these deformations:

$$D_{m,i} = k_i * u\left(< \vec{P}_{vertex\_i} - \vec{P}_{HM\_m}, \vec{N}_{vertex\_i} > \right) \quad \text{where}$$

$$k_i = 1 - (d_{m,i} / radius\ of\ influence)^2,$$

$$d_{m,i} = \left\| \vec{P}_{HM\_m} - \vec{P}_{vertex\_i} \right\|.$$

The deformations from different points of the *haptic interaction mesh* are not summed up but a maximum value is calculated to obtain vertex displacement:

$$D_i = \max_m \left(D_{m,i} \right)$$

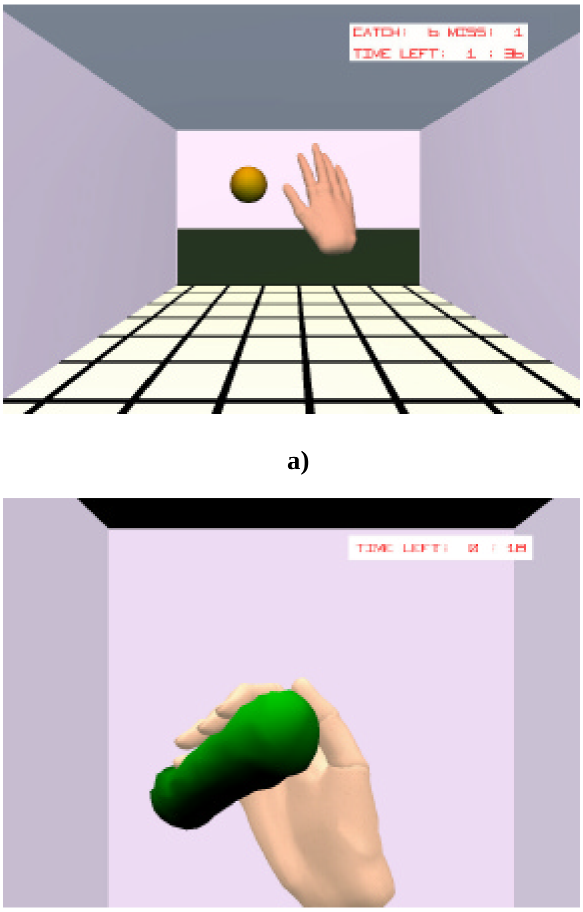
Figure 4 presents a sequence of deformations while squeezing a ball. The screen shoots were taken while the hand was moving.

4. Applications

The VR simulations described here were developed using a commercial graphics library [10]. All simulations contain a high-resolution virtual hand [14] and deformable objects (rubber ball, power putty). Several hand gestures allow users to interact with the virtual objects: whole hand grasping, two finger grasping (lateral pinch) and selecting (pointing with the index finger). Contact detection is checked between the hand and the virtual objects and triggers the grasping gesture. Objects stay “attached” to the virtual hand until a release gesture is executed.

The essential feature of these simulations is the user's interactivity with the VE. Each simulation has several parameters that can be selected by the user, such as level of difficulty of the rehabilitation exercise, the time allowed to complete the exercise, etc..

The first simulation example implements a rubber ball squeezing exercise [9]. The ball stiffness is color-coded and can be selected by the patient at the beginning of the rehabilitation routine. Ball dynamics simulate gravity and Newtonian laws. Once the ball is grasped, it deforms in contact with the virtual hand while force feedback is felt by the patient. The force calculation uses the local haptic loop and the rendering model described previously. A mesh composed of five points (mesh center and four first-degree neighbors) was used for interaction.



a)

b)

Figure 5: Virtual simulations: a) Ball Game; b) Power Putty molding [9].

A more elaborate simulation is the Ball Game shown in Figure 5-a [9]. The patient has to throw the ball so that it hits the target wall above a marked area. When the ball bounces back the patient has to catch it after at most one bounce off the ground. The ball speed parameter ("fast" or "slow") is selected at the beginning of the simulation. Any correct catch increases the patient's "catch" counter while a miss will increase the "miss" counter. The ball deforms when caught by the user and loses energy while bouncing.

The third simulation example is a molding of virtual "power putty," as illustrates in Figure 5-b. The patient selects between an ellipsoid or sphere unmolded putty shapes, each with three selectable hardness levels. Plastic deformation and distributed haptic rendering models were used for this simulation. The ellipsoidal putty is used for full grip exercise. The spherical putty is used for finger pinch exercise.

5. Conclusion and Future Research

Modeling deformations and haptic interactions of complex shaped objects needs more elaborate methods than previous HIP-based interaction models. A new method for modeling virtual hand haptic interactions was proposed. The model uses a *haptic interaction mesh* to calculate haptic interactions and object deformations. This model allows better quantification of local haptic interactions. The method is not dependent on the haptic interface hardware used. However, the mesh parameters need to be customized according to the geometry of the haptic device representation in VE used for the interaction. More research is needed for mesh point placement, dynamic mesh control and force mapping. Results on force pattern distribution in grasping [11] can be combined with this method to generate more realistic grasping interactions.

The haptic rendering method was applied for modeling the interactions of the RM-II interface in a medical application (orthopedic rehabilitation). Haptic interaction meshes were defined at each fingertip and used in the force calculation and virtual object deformation. The method was used to enhance real-time simulations that involve elastic and plastic deformations and physical modeling (dynamics, reflection law, and gravity). The experimental system described here uses only a haptic glove. Elbow and knee haptic interfaces are currently being designed and will be integrated in VR simulations. The haptic rendering techniques described above will be applied for the modeling of these new haptic interfaces.

6. Acknowledgments

Work reported here was supported by grants from the National Science Foundation (grant BES-9708020) and from the Rutgers University (with funds provided by SROA and the CAIP Center).

7. References

[1] Basdogan, C., Ho, C., Srinivasan M.A., (1997), "A Ray-Based Haptic Rendering Technique for Displaying Shape and Texture of 3D rigid objects in Virtual Environments", the Winter Annual Meeting of ASME'97, Nov. 15-21, Dallas, TX, DSC-Vol. 61, pp. 77-84.

[2] Bouzit M., (1996), "Design, Implementation and Testing of a Data Glove with Force Feedback for Virtual and Real Objects Telemanipulation," PhD. Thesis, Paris, France.

[3] Burdea G., (1996), *Force and Touch Feedback for Virtual Reality*, John Wiley & Sons, New York, NY.

[4] Gomez D., G. Burdea and N. Langrana, (1995), "Integration of the Rutgers Master II in a Virtual Reality Simulation," *IEEE Virtual Reality Annual International Symposium*, pp. 198-202.

[5] Foley, J., van Dam, A., Feiner, S., & Hughes, J. (1990), *Computer Graphics: Principles and Practice* Second Edition. Addison-Wesley Publishing Company, Inc.

[6] Ho, C., Basdogan C., Srinivasan, M., (1997), "Haptic Rendering: Point- and Ray-Based Interactions," *Proceedings of the Second PHANToM Users Group Workshop*, Dedham, MA, Oct. 20-21.

[7] Massie T., Salisbury, J., (1994), "The PHANTOM haptic interface: a device for probing virtual objects," *Proceedings of the ASME Dynamic Systems and Control Division*, DSC-Vol. 55-1, Chicago, IL, pp. 295-301.

[8] Patounakis, G., Bouzit, M. & Burdea, G. (1998), "Study of the Electromechanical Bandwidth of the Rutgers Master," Technical Report CAIP-TR-225, Rutgers University, May 22.

[9] Popescu, G., G. Burdea, M. Bouzit and V. Hentz, (1998), "Orthopedic Telerehabilitation with Virtual Force Feedback," *Presence,* MIT Press, submitted.

[10] Sense8 Co. (1994), WorldToolKit User's Manual, Sausalito, CA.

[11] Shimizu S., Shimojo M., Sato S., Seki Y., Takashi A., Inukai Y., Yoshioka M., (1996), "The Relation between Human Grip Types and Force Distribution Pattern in Grasping", 5th IEEE International Workshop on Robot and Human Communication, pp. 286-291.

[12] Srinivasan, M., Basdogan, C., (1997), "Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges," *Computer & Graphics*, Vol. 21, No 4., pp 393-404.

[13] Turner M., Gomez D., Tremblay M., Cutkovsky M., (1998), "Preliminary Tests of an Arm-Grounded Haptic Feedback Device in Telemanipulation," *Winter Annual Meeting of ASME'98,* Nov. 15-21, Dallas, TX, DSC-Vol. 64, pp. 145-149.

[14] Viewpoint Datalabs. (1993). Catalog of 3D Models, Orem, UT.

[15] Zilles C.,and Salisbury K., (1995), "A Constraint-based God-object Method For Haptic Display", *IEEE International Conference on Intelligent Robots and Systems '95*, Pittsburgh, PA, Vol. 3, pp. 146-151.