HAPTIC RENDERING: PROGRAMMING TOUCH INTERACTION WITH VIRTUAL OBJECTS

K. Salisbury[*] D. Brock[†] T. Massie[‡], N. Swarup[‡] C. Zilles[‡]
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

1 Abstract

Haptic rendering is the process of computing and generating forces in response to user interactions with virtual objects. Recent efforts by our team at MIT's AI laboratory have resulted in the development of haptic interface devices and algorithms for generating the forces of interaction with virtual objects. This paper focuses on the software techniques needed to generate sensations of contact interaction and material properties. In particular, the techniques we describe are appropriate for use with the Phantom haptic interface, a force generating display device developed in our laboratory. We also briefly describe a technique for representing and rendering the feel of arbitrary polyhedral shapes and address issues related to rendering the feel of non-homogeneous materials. A number of demonstrations of simple haptic tasks which combine our rendering techniques are also described.

2 Introduction

The process of mechanically interacting with with remote and virtual objects has been of interest to researchers for a long time. Handling of distantly located objects through remotely controlled manipulators has been feasible since at least the early days of handling hazardous nuclear materials [12]. In these systems a master control device is used to control the actions of the remote manipulator. "Force reflection" is sometimes used to present to the user, through the master, forces encountered by the remote manipulator. This permits perception and manipulation of these remotely located objects. In the early 70s researchers began to simulate this type of interaction through the use of simple mechanical models of objects in the environment. By computing the forces which would be encountered in interactions with real objects and displaying them through a force reflecting interface, the sensation of touching objects could be created [10]. These "haptic" interactions with simulated objects represent one of the first instances of mechanical interaction with virtual objects.

Haptic interactions have been used to aid investigations of molecular docking [1]. This task requires the user to follow a force gradient until the molecules are interlocked. The force field the molecules move through is derived from models of inter-molecular forces. Although a realistic calculation of these forces is computationally intensive, they can be applied to the user as simple attractions or repulsions and used to find suitable docking configurations. This approach has been found to be useful for this complex, molecular level, task.

The molecular docking task does not, however, require generation of the same type of contact forces that we encounter in everyday manipulation of the objects. Forces resulting from contact, palpation, and stroking actions require generation of macroscopic forces which give rise to sensations of shape, surface hardness, texture and friction. Kilpatrick [5] found it suitable to model hard surface interactions using Hooke's law augmented with clicks when virtual contact is made. He recommended, in addition, a mechanical brake making surfaces "feel" harder, to "radically increase friction when a virtual hard surface is encountered."

Recent interest in creating and interacting with virtual environments (VEs) has begun to push these ideas to new levels of sophistication. Taking advantage of advances in graphic display, computational capability and modeling of visual representation has permitted the visual component of complex virtual environments to be rendered

---

[*] Principal Research Scientist, Dept. of Mechanical Engineering
[†] Research Scientist, Artificial Intelligence Laboratory
[‡] Graduate Student, Dept. of Mechanical Engineering

1995 Symposium on Interactive 3D Graphics, Monterey CA USA

with good fidelity. The ability to perform mechanical or "haptic" interaction with these scenes has lagged significantly behind. The majority of VE systems in use today rely on passive devices, such as instrumented gloves and joysticks, to track user motions and permit limited interaction with virtual objects. To provide force feedback to users in these systems, a few researchers have adapted teleoperator masters and in some instances have developed dedicated haptic feedback devices. Advances in haptic interaction have been limited due to lack of high performance interface devices, and the lack of a coherent approach to object modeling and sensory display of mechanical attributes.

Our work at at MIT has begun to address this problem with integrated investigations into the science of haptics. The term haptics has come to be used by the VE and telerobotics communities to refer to the sensorimotor interactions which occur during perception and manipulation of mechanical objects. We have concentrated on methods for tracking the motion of the human finger and applying precisely controlled forces to the user's fingertip through a ground-based haptic interface, the Phantom [6]. A wide range of demonstrations have shown that our device has sufficiently clean dynamics (stiff, low-friction, backdrivable) to display a wide dynamic range of impedances with high fidelity [2]. As a result of the high sampling rate, sensor resolution, and structural stiffness of our haptic interface, the dynamic modes of the haptic interface are highly decoupled from the programmed dynamics of the virtual environment. Thus, transparency to the dynamics of the interface hardware is achieved and representation of the virtual environment dynamics is greatly facilitated. As a simplifying assumption, we have focused on point interactions. Point contacts with objects permit only pure forces (no torques) to be exerted through the contact and require only three active (powered) motions in the haptic interface to faithfully reproduce the force geometry. The minimal complexity of the system has helped achieve good bandwidth by reducing parasitic structural and actuator mass. This reflects our view that good temporal display quality is at least as important as good spatial characteristics in a haptic display. The point paradigm is not a particularly restrictive assumption in that multiple points can be combined to exert torques on objects and control their orientations as with human fingertips.

While researchers have begun to look at algorithms for generating forces resulting from contact with virtual objects [3,11], we feel that there is a great need for a more coherent approach to generating (or rendering) these sensations and modeling interactions with complex objects. Our interest is in developing a framework in which we may represent shape, surface properties, bulk properties and multiple object interactions. Such a framework should permit the representation of a wide variety of objects and object interactions, while simultaneously addressing the problems of real-time generation of appropriate sensations. We can view haptic interactions as really occuring at two levels. When a contact occurs there is a net force (vector) experienced (or generated) by the user. In addition the distribution of the forces (or tractions) which occur at each contact site are perceived through the user's mechanoreceptors, giving rise to our tactile sense. Because of the difficulty in building tactile displays which present the spatial distribution of forces at each contact, we have focused on displays which present only the net force information. We have found that if this force information is presented with sufficient bandwidth and resolution, many effects that we consider to be tactile sensations can be created. Surface shape, compliance, texture and friction can all successfully be evoked through proper modulation of the net force exerted on the human. A general framework for haptic rendering must then be able to represent and permit display of these and other basic haptic elements. These elements, in turn, must be contained in a larger framework which represents the object shape and bulk properties appropriate for rendering the larger scope of interactions that occurs during object motion and inter-object interactions.

Though our efforts to build a general haptic rendering system are still in the early stages, we have made progress in the rendering of basic contact interaction elements, macroscopic object shape properties, and bulk object properties. We describe below our progress in these areas.

3 Rendering Haptic Elements

A region in space populated with objects can be divided into volumes which represent free space and volumes which represent objects. The surfaces of these objects comprise the boundaries between the two. Perception of the shape and details of these objects is accomplished by haptic exploration in which these surfaces are palpated and stroked. We discuss below the various haptic elements which must be available to enable active exploration and perception of objects, many of which we have implemented. Taken together, these elements permit higher level tasks such as grasping and manipulation, some of which we have demonstrated.

3.1 Freespace Movement

A haptic rendering system must first be able to give the sensation of free space. To do this requires a haptic interface with intrinsic characteristics that allow it to be effortlessly moved about, with little distraction from mechanism friction, inertia and vibration. In using the Phantom interface this is enabled by the device's intrinsically low backdrive friction and inertia. In addition, the mechanism's smooth transmission characteristics and well damped high natural frequency reduce unintended vibrations to nearly below perceptible levels.

3.2 Contact Transients

At the instant of contact with a surface rapid onset of force occurs with sufficient impulse to remove momentum from the user's finger or tool. This requires good bandwidth and stiffness in the interface to provide quick stable, onset of force. We and others typically accomplished this by programming a one-sided spring function to generate repelling forces that increase with surface penetration. As discussed below, careful control of this contact impedance can be used to vary and enhance perceived material properties.

3.3 Contact Persistence

The sensation of sustained contact with a surface requires that the user be able to push into it and experience compressive contact forces of sufficient magnitude to make it feel solid without actuator saturation or instability. We have found that it is not necessary to generate huge forces to create the illusion of solid immovable walls. In fact, when performing manipulation involving motion at only the elbow, wrist, and fingers, users rarely exert more than 10 Newtons of force. The illusion of solid surfaces, is reinforced by the contrast between these contact forces and the low free-space forces imposed by the Phantom (typically less than 0.1 Newtons).

3.4 Contact Impedance

While not completely separable, we can divide the impedance of an object into two components, the local or contact impedance and the net or gross impedance of the object. The contact impedance gives rise to sensations of material properties. As other researchers have recognized [3], we have found that adding viscous damping to the characteristic equation for a constraint surface greatly enhances the user's perception of a hard surface. Perceptually, a wall simulated in our system by $f = Kx + Bv$ can be made to feel like hard plastic, whereas a wall simulated by $f = Kx$, using the same value for K, would feel spongier than a typical mouse-pad. Effectively, adding a damping term will change the coefficient of restitution between a user and the virtual surface [11].

3.5 Frictionless Surfaces

When a user only experiences forces normal to the surface being touched the sensation of a slippery or frictionless surface is evoked. Computing contact forces in this case requires only the determination surface normals and penetration depth. This is, in fact, is the easiest haptic effect to generate with our system. The same good intrinsic properties of the Phantom system which permit the sensation of free space motion contribute to the faithful rendering of frictionless motion in the 2 dimensional subspace of sliding across a surface. While using the Phantom to touch friction-free surfaces, users have described the sensation as that of "an ice cube sliding on glass."

3.6 Surface Friction

Imposing tangential forces on users while they stroke a surface adds an important sense of realness to perception of objects. In real life, we rarely experience frictionless surfaces and, in fact, heavily rely on friction in tasks involving manipulation. We have developed several techniques which approximate both stiction and Coulomb friction (static and dynamic friction). As with [11], we recognize the importance of incorporating static friction into the friction model. In our implementation the model has two states: sticking and sliding. When contact is first made we store the location of contact and begin the stiction state. If the user tries to slide along the surface, tangential forces (using Hooke's law or impedance control) are applied to restore the user back to his initial point of contact, the "stiction point". If the force required exceeds the normal force times the the coefficient of friction, then we change to the sliding state.

Unlike [11] we model the sliding state with Coulomb friction rather than simple viscosity. Coulomb friction involves applying a retarding force which is only a function of the coefficient of friction and normal force, in the direction opposite to the direction of motion. Due to the difficulty in accurately measuring small velocities in a sampled data system, we designed a robust method which requires only position measurements. When transition to the sliding state occurs, we know the displacement from the stiction point and can assume the user is moving in the direction of this displacement. To create a tangential force with the correct magnitude and direction we simply need to move the stiction point to a new place on the line which connects the user and the old stiction point. The stiction point's offset from the haptic interface point can be calculated by dividing the friction force (the normal force times the coefficient of dynamic friction) by the stiffness. Once the new stiction point is assigned we return to the stiction state.

By setting the coefficient of dynamic friction below the coefficient of static friction we have demonstrated a convincing stick-slip sensation. The vibration generated during object motion against friction modeled in this way evokes a sensation of slippage. In the `BLOCKS` demonstration program (program images shown at end of paper) a user is able to pick up a virtual cube with two Phantoms; if the objects slips, the user can detect this occurrence by attending to these vibration and net force direction cues. Without such a friction model, force closure grasps of virtual objects would not be possible. By using friction to enable grasps the `BLOCKS` program permits the blocks to be stacked, thrown, dribbled, and juggled [15].

3.7 Surface Curvature

Surface discontinuities at edges and corners are primarily perceived in humans by mechanoreceptors sensitive to curvature. However we have demonstrated that these basic curvature sensations can be convincingly be displayed by control of the normal force vector. Users will perceive a discontinuity of the normal direction as an edge or corner; one key to making smooth objects is to vary the direction of the force vector continuously. By utilizing surface normals at the vertices (defined say, by averaging ajacent facet normals), a satisfying normal force direction can be found at any point via interpolation between these vertex normals (much like Phong shading in graphics).

We have found the actual shape of an object to be rather insignificant in making objects feel smooth. Because of the inaccurate position sense that humans have, a coarsely meshed polyhedron will be perceived as smooth if a suitable surface normal interpolation scheme is used. This has been demonstrated in a pair of example programs we have written. One program, models a surface by assigning heights on a 2-D mesh. Complex surfaces including a telephone and a baboon's face have been "rendered" by interpolating height and surface normal between points in this matrix of heights. In the case of the phone rendering actual heights were measured and entered into the mesh. In the case of the baboon face, a pseudo-height map was derived from image point brightness. Though this does not really represent the true shape, it provides sensations of underlying geometry. The second program, (`SMOOTH`) presents the user with a smoothed rendering of a polyhedrially modeled asteroid shape. It is rendered using the constraint-based god object method described below, with the addition smooth interpolation of surface normals across edges. The result is that the previously sharp edges feel rounded.

3.8 Surface Texture

The sensation of texture results from both the effects of small shape details and friction on surfaces. In direct manipulation humans can utilize both their tactile sense (fingertip mechanoreceptors) and net force sense to perceive texture. Conveniently (since we currently lack good tactile array force displays), variations in net force applied to a user can generate texture sensations. Minsky [8] presented users with variations in tangential forces dependent on local shape variations to evoke a wide variety of texture sensations. We have also used shape-driven variations in normal force to evoke sensations of texture on a frictionless surface [6]. To be complete, variations in normal and tangential forces should be used together to simulate texture with force-based displays. The stick-slip sensation demonstrated by [15] does address part of this need in providing a purely friction dependent sense of texture. It remains to combine both shape and friction dependent force variations to display more complex texture.

We have begun to explore techniques similar to graphics texture mapping that can be used to overlay the surfaces of objects with standard textures. For example, one could define a texture map which induces slight reorientations in the rendered contact normal of a surface facet to which it is applied. Making this perturbation a function of location on the facet reflects the spatial dependence of texture, however care must be taken to not alter the spatial frequency when the texture is mapped to facets of different scale.

We have also created convincing shape dependent textures by using a height-map function applied to planar surfaces. At every point on the planar surface, the software calculates a height offset and a normal vector offset, as defined by the height-map. The texture patch is defined by a grid of heights, and is constructed to permit tiling on a bigger surface without texture discontinuity between adjacent patches. For example a suitable continuous texture patch can be defined by assigning $z = \cos(x)\cos(y)$ with x and y in the range $(0, 2\pi)$. It is interesting to note that depending on the period and amplitude of such a texture, users may perceive it as a shape, as a texture, or in the limit, as friction.

3.9 Net Object Motion

In the preceding sections, we have primarily addressed local effects in which little or no net object motion occurs. In fact, some of our efforts have investigated interaction with objects that are free to move in one or more dimensions. By tracking contact forces according to the above techniques and applying these forces to a model of the object's mass, stiffness and viscosity with respect to ground, it is relatively easy (in few-degree-of-freedom systems) to integrate the resulting accelerations and compute net object displacements. Demonstrations of spring centered switches (`SLIDERS`) and switches with detents (`BUTTONS`) have been made and suggest a rich range of virtual controls which may be constructed. A demonstration which permits pushing of masses on a frictionless surface (`MULTY3`) shows the ability to interact with dynamic objects and control two-degree-of-freedom motions. Two phantoms have been used together in a program (`BLOCKS`) which permits grasping and placing cubes which are free to move in rectilinear (three-degree-of-freedom) motion. Extending these capabilities to full six-degree-of-freedom motion including manipulation and assembly tasks is clearly a formidable undertaking but one which requires a firm understanding of the local effects we have addressed to date. Significant extensions are required to address the kinematics of articulated objects such as mechanisms and objects with transient kinematics, such as are encountered during assembly and tool interaction.

4 Shape Representation

It is desirable to not only display local surface properties, but also overall shape of objects. We have implemented a number of techniques to describe shape. An evolution of techniques is in progress, starting with vector field implementations, progressing to god object representations, and looking ahead to potential energy function representations.

Our vector field methods subdivide the volume of an object and associate a sub-volume with each surface. When the haptic interface is in a sub-volume, a force whose magnitude is a function of the distance penetrated is applied in the direction of the normal to the associated surface[6]. These vector field methods conceptually create a map of the 3-D object volume and assign a force vector to each location, so that during each servo loop the contact force can be looked up.

This method works rather well for simple geometric shapes because it is reasonably easy to construct these subspaces by hand. For planes aligned with the coordinate axes the force vector can be computed from a simple $F_x = Kx$ relation. For spheres, the direction is that of the vector pointing from the sphere's center to the haptic interfaces endpoint, and the magnitude is the distance the endpoint has penetrated the sphere's surface scaled by a constant. The simplicity of this method has allowed us to explore many aspects of haptic rendering, but it has its drawbacks. When designing more complex objects it is less obvious how to sub-divide the volume, and thin objects are susceptible to being pushed through.

The central difficulty is that the maximum stiffness of any virtual object is limited, due to the inherent mechanical compliance of haptic interface devices. This means that the user's contact point often penetrates simulated object volumes to a greater distance than would be possible in real life, leading to an ambiguity in determining which surface was entered. A better method was needed to keep track of the surface being stroked if believable forces were to be displayed robustly.

The constraint-based god object method employs a strategy to stop the haptic interface's virtual contact point from penetrating objects[14]. By concentrating on surfaces rather than volumes, we attempt to more realistically compute forces, and incidentally give ourselves access to an enormous body of objects already in existence in standard surface representations. This method keeps track of a virtual contact point (the god object) which remains on the surface when a virtual object is probed. With the location of the god object on the surface, there is no ambiguity in which force vector should be applied to the user.

Given the previous location of the god object and the current location of the haptic interface, the algorithm will identify a number of surfaces on the rendered object which are currently involved in the interaction and denote them as active. A surface is *active* if the god object is on one side of the rendered surface, and the haptic interface is on the other, and the action takes place within the boundaries of the surface. One surface can be active for each powered degree of freedom in the device.

Once this set of surfaces, or constraints, has been identified the new location of the god object can be computed. By finding the closest point on the active constraint surface to the current haptic interface point we can determine the new location of the god object (strictly, this applies to the frictionless case, but can be extended to include surfaces with friction. Since we chose planar constraints, the solution can be found by solving a set of linear equations requiring only 65 multiply or divide operations to calculate the coordinates.

This method will create a faceted object which can exhibit sharp corners; smoothed objects can also be rendered by adding a smoothing algorithm. We are currently in the process of combining the basic effects described above witht the god object renderer. We expect this to result in a fairly rich system in which arbitrarily shaped polyhedral objects may be rendered with controllable degrees of smoothing, friction, surface impedance. In the next section we address another approach to rendering complex shapes which lends itself to rendering objects with bulk material properties which are significantly non-homogeneous.

5 Rendering Non-homogeneous Materials

Although the methods described above permit a large class of objects to be rendered, they do not directly address objects composed of non-homogeneous materials. Incorporation and presentation of non-homogeneity greatly extends the class of objects that can be presented, particularly tissue surrounding the internal organs and skeletal structure of vertebrates. For instance, haptic presentation of biological objects will be an integral component in multiple modality surgical environment simulations. We describe below preliminary work which concentrates on local surface impedance properties [13].

We are concurrently developing approaches to the haptic scanning of surface property data based on force sensing, analogous to the visual scanning of pictures to produce image data. We envision mechanically probing an object at discrete surface points, capturing local surface properties through force and position measurements, and finally storing the data in a format readable by the haptic renderer. Due to the inherent sampling nature of scanning, the haptic rendering of the sampled surface data must be able to sufficiently reconstruct the original surface properties without perceptual loss of information. Hence, the techniques we use to haptically *represent* surface information are intrinsically coupled to the issues involved in haptically *scanning* surface properties.

In contrast to computer graphics which involves global environment rendering, haptics primarily involves local interactions. For a large class of objects, local interactions are decoupled from global object dynamics. Consequently, efficient computational haptic rendering algorithms should take advantage of this local nature. As the user moves their interaction point on the surface, the haptic renderer will only render the local "window" of surface representation data about that point. We have successfully demonstrated haptic rendering of non-homogeneous objects by employing this haptic window technique.

5.1 Rendering Methods

The geometric modeling technique of B-spline surfaces is utilized to interpolate discrete, spatially distributed, values of surface impedance data. B-splines are particularly appropriate to haptic rendering because they are comprised of a set of blending functions that has only local influence and are dependent on a finite number of neighboring control points[9]. Furthermore, the order of the interpolating polynomial is not affected by the number of control points. Both of these facets complement the attributes of the haptic window which only renders local properties. To ensure smooth haptic transitions across non-homogeneous sample points, C^2 continuity is imposed on the B-spline which results in cubic interpolating surfaces. As a result, a 4 x 4 patch of data points is necessary to construct the interpolation polynomial.

Although geometric interpolation of surface impedances provides an efficient and simple means for rendering surface properties, there are limitations. Primarily, geometric interpolation does not guarantee that a closed circuit interaction path with the virtual object will be conservative, hence potentially providing the sensation of an unrealistic "active" surface. It is possible to interact with the surface in a compliant area expending little work, move tangentially to an area of higher impedance, and then leave the surface with nonzero net energy transfer. In order to ensure passivity of the surface, requirements must be placed on the internal force field and boundary conditions imposed by the surface. Specifically, if the force field $\mathbf{F}$ within the surface can be described as the negative gradient of a scalar potential field Φ,

$$\mathbf{F} = -\nabla\Phi \tag{1}$$

and the potential at the surface is constrained to be constant everywhere, then *any* closed path interaction with the object will be conservative.

We are investigating potential field methods which respect the passivity requirements directly. Conceptually, we can use static electro-magnetic fields to model object properties. Representing surfaces as perfect conductors permits us to enforce equal potential at entry and exit from touching an object. Solutions of Laplace and Poisson equations, can then be used to solve for the value of forces at points internal to the object. If we then wish to set the local impedance at points within the object, we may impose further internal boundary conditions on the potential field. Thus, we may conveniently map impedances measured for real objects into the geometric model of the object's force generating function using the above relationship.

6 Haptic Demonstrations

A number of demonstrations (illustrated in screen images shown below) have been developed which use the basic haptic elements, described in the previous section, as building blocks for more complex applications. Initially simple geometric shapes, such as spheres, cubes, and polyhedra, were constructed and implemented using the vector force field approach described earlier. Dynamic objects, illustrated in Figure 1, were later developed. These simulations allowed users to push and slide objects, permitting the discrimination of virtual mass and inertia. Surface effects between objects, including stiction and Coulomb friction were also added.

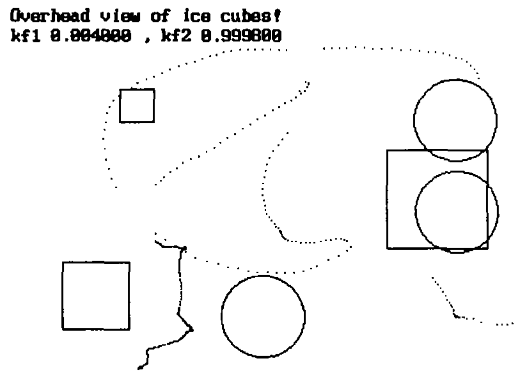

Figure 1: `MULTY3`, a dynamic simulation which allow users to push and slide virtual objects, and permit the discrimination of mass, inertia, friction, and impact.

An application which became immediately apparent, was the virtual control panel. Knobs, buttons, sliders, and switches, featuring clicks, detents, toggles, and stiffnesses, allowed users to "feel" and operate virtual instruments, as shown in Figure 2. Another application, which may have significant importance, is the simulation and rehearsal of medical procedures. Figure 3, shows the screen image of the needle biopsy simulator we developed. A magnetic resonance image (MRI), acquired from the Brigham and Women's Hospital, was segmented along a user specified line. Mechanical properties including stiffness, tear strength, and viscous friction were assigned to each layer, so that the surgeon could feel the pressure of needle against the tissue and "pop" as each layer was pierced.

Using a distributed interactive simulation approach, we created a tissue palpation demonstration. The haptic device, controlled by a 486PC, transmitted probe and tissue information via the network to a SGI Indigo^2 Extreme. Thus the user could feel the compliant surface while viewing a high-quality graphics image, as shown in Figure 4.

Using standard graphics file formats, we were able to haptically render arbitrary convex and concave objects. Figure 5 shows an "asteroid" imported from a `.plg` file and presented to the user to both push and probe. This example is prepatory to the development of standard object interchange format which will allow visual, haptic, acoustic, and functional representation. Finally, combining the ob-

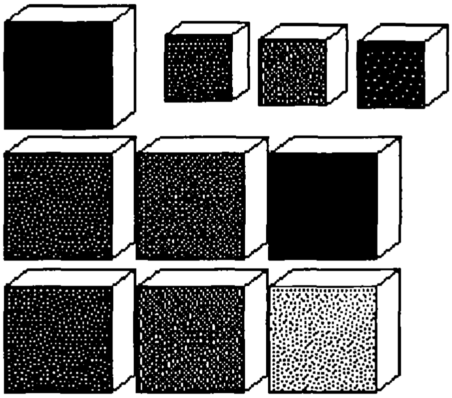

Figure 2: Virtual instrument panels include knobs, buttons, sliders, and switches, with clicks, detents, toggles, and stiffness. Illustrated is `BUTTONS` program.

Figure 3: A needle biopsy simulator demonstration, `MRI`,allowed surgeons to experience the sensation of pressure of the needle against tissue, and the "pop" as each layer is piered.

Figure 4: `DEFORM`, a tissue palpation demonstration using the Phantom haptic device running on a PC and a Silicon Graphics workstation to provide graphics support.

jects, elements, and algorithms developed above, we build a simple "virtual world" composed of building blocks and virtual fingertips to manipulate them. The user employed two haptic devices to pick up and toss the cubes around the room, while feeling the friction, mass, inertia, and impact of these objects.

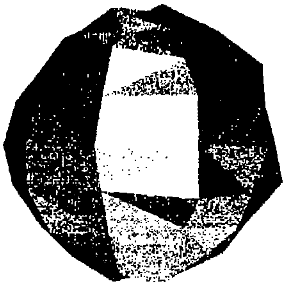

Figure 5: Standard graphics file formats were imported and rendered with the **ASTEROID** program. The program permits palpation and exploration of interior or exterior surfaces.

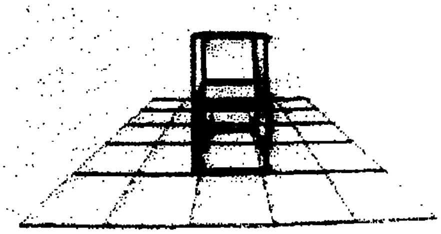

Figure 6: **BLOCKS**, a program which renders two blocks that can be grasped and manipulated using two fingertips.

7 Acknowledgments

This work is supported in part by NAWCTSD contracts, N61339-93-C-0108, N61339-93-C-0083, N61339-94-C-0087, ONR Grant N00014-93-1-1399, fellowships from the National Science Foundation Graduate Fellowship program and the National Defense Science and Engineering Fellowship program.

8 References

1. Brooks, Jr., Fred P., M. Ouh-Young, J. J. Batter, and P. J. Kilpatrick. "Project GROPE – Haptic Displays for Scientific Visualization." Proceedings of SIGGRAPH '90. Dallas, Texas. August 6-10, 1993.

2. Colgate, J. Edward, and J. Michael Brown "Factors Affecting the Z-Width of a Haptic Interface." Proceedings of the IEEE International Conference on Robotics and Automation, 3205-3210. 1994.

3. Colgate, J. Edward, P.E. Grafing, M.C. Stanley, and G. Schenkel. "Implementation of Stiff Virtual Walls in Force-Reflecting Interfaces." Proc. IEEE-VRAIS, pp. 202-208, 1993.

4. Durlach, Nathaniel I., et al. "Virtual Reality: Scientific and Technological Challenges." Report produced for the National Research Council, National Academy of Sciences. Washington D.C. December 1994.

5. Kilpatrick, P. J. "The Use of Kinestetic Supplement in an Interactive System." Ph.D. dissertation, Computer Science Department, University of North Carolina at Chapel Hill. 1976.

6. Massie, Thomas H. "Design of a Three Degree of Freedom Force-Reflecting Haptic Interface." SB thesis, MIT EECS Department. May, 1993.

7. Massie, Thomas H. and Kenneth Salisbury. "The PHANToM Haptic Interface: A Device for Probing Virtual Objects." Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. Chicago, IL. November 1994.

8. Minsky, Margaret M., et al. "Feeling and Seeing: Issues in Force Display." Computer Graphics, vol. 24, no. 2, pp. 235-243, 1990.

9. Mortenson, Michael E. Geometric Modelling. New York: John Wiley & Sons, Inc. 1985.

10. Noll, A. Michael. "Man-Machine Tactile Communication." Society for Information Display Journal. July/August 1972. Reprinted in Creative Computing, July/August 1978 p.52-57.

11. Salcudean, S. E. and T. D. Vlaar. "On the Emulation of Stiff Walls and Static Friction with a Magnetically Levitated Input/Output Device." Proceedings of the ASME Dynamic Systems and Control Division. Chicago, IL. Nov. 6-11, 1994.

12. Sheridan, Thomas B. Telerobotics, Automation, and Supervisory Control. Cambridge, MA: MIT Press, 1992.

13. Swarup, Nitish. SM thesis in progress. Department of Mechanical Engineering, MIT. Expected May 95.

14. Zilles, Craig and Kenneth Salisbury. "A Constraint-Based God Object Method for Haptic Display." Submitted to IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots. 1995.

15. Zilles, Craig. SM thesis in progress. Department of Mechanical Engineering, MIT. Expected May 95.