Initial Haptic Explorations with the Phantom: Virtual Touch Through Point Interaction

by

Thomas Harold Massie

B.S., Electrical Engineering
Massachusetts Institute of Technology, 1993

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1996

Signature of Author...
Department of Mechanical Engineering
January 19, 1996

Certified by ....
J. Kenneth Salisbury, Jr.
Principal Research Scientist
rvisor

Accepted by..
Ain A. Sonin
Chairman, Department Committee on Graduate Students

Initial Haptic Explorations with the Phantom: Virtual Touch Through Point Interaction

by

Thomas H. Massie

Submitted to the Department of Mechanical Engineering
on January 19, 1996 in partial fulfillment of the requirements for the
Degree of Master of Science

ABSTRACT

The primary topic of this research was the implementation and programming of a force reflecting haptic interface, known as the PHANToM (Personal Haptic iNTerface Mechanism). The goal was to develop software models that would allow users to feel and manipulate the data represented within computers. Compact models of texture, shape, compliance, viscosity, friction, and deformation were implemented using a point force paradigm of haptic interaction. The motivation for and implications of a point force haptic interface are discussed in detail. Finally, for those who are not immediately convinced of the utility of enabling humans to interact with computers through the sense of touch, a few of the applications enable by the PHANToM are described.

Thesis Supervisor: J. Kenneth Salisbury, Jr.
Title: Principal Research Scientist

Table of Contents

1. Introduction

This thesis addresses the design, programming, and implementation of a haptic interface designed solely with the intent of facilitating human-computer interaction. The Personal Haptic iNTerface Mechanism (PHANToM) uses a novel thimble-gimbal to achieve a "point force approach" to haptic interaction. Chapter 2 is dedicated to describing this approach as it has many implications for human perception and software development. Chapter 3 addresses the basic software issues, and Chapter 4 describes the applications for this particular haptic interface. Finally, Chapter 5, describes why point force haptic interfaces like the PHANToM will soon see wide use.

1.1 Haptic Interfaces

Real-time, photo-realistic 3-D computer graphics, and even spatialized 3-D sound are now achievable on high-end computer workstations. At the current pace of computer technology, few would argue that computer generated graphics and sounds, virtually indistinguishable from the real world they mimic, will be available on the personal computer within a few years. Visual and auditory feedback alone cannot enable a person to interact with the computer as naturally as he would interact with his real environment.

A significant component of our ability to visualize, remember, and establish cognitive models of the physical structure of our environment stems from haptic interactions with objects in the environment. Kinesthetic, force and cutaneous senses combined with motor capabilities permit us to probe, perceive and rearrange objects in the physical world. Information about how an object moves in response to applied force and the forces which arise when we attempt to move objects can provide cues to geometry (shape, locality, identity), attributes (constraint, impedance, friction, texture) and events (constraint change, contact, slip) in the environment [17].

In the course of a typical human computer interaction, a user views output on a video monitor, and works to change the input via a mouse, joystick, or keyboard. In general, human beings do not interact with the world this way. Rather we use our hands to both **change** *and* **measure** our environment. Imagine an artist molding clay, a jeweler fixing a watch, or a surgeon searching for a bullet through a small incision. These activities involve such an immediate level of manual interaction that they are not easily described in terms of "input" and "output". Unlike other sensory modalities, haptic interactions permit two-way interaction via work exchange. Controlled work can be performed on dynamic objects in the environment and modulated to accomplish tasks. Today, haptic feedback is woefully missing from the typical human computer interaction.

In that it was designed solely for human-computer interaction, the mouse is a unique exception among the list of human-computer interfaces. Consider the keyboard (typewriter), the cathode ray tube (television), the joy-stick (rate controller), and audio speakers (telephone receiver). All of these devices existed long before computers were attached to them, and arguably, only incremental improvements to these interfaces have been achieved. Haptic interfaces pre-date computers as well. Force-reflecting teleoperators were used to give humans a sense of presence in remote or dangerous environments long before they were attached to computers. The first computerized haptic interfaces were teleoperators, interfaced to computers which simulated the remote environment virtually [3]. In fact, some of the commercially available haptic interfaces today like the Sarcos Arm [25] and the Cybernet Per-Force [7] are teleoperator master controllers adapted for use in virtual environments.

In contrast to these earlier haptic interfaces, and analogous to the development of the mouse, haptic hardware and software algorithms with the specific goal of improving human-computer interaction are now under development. Some of the first efforts to build hardware designed specifically for enabling people to touch virtual environments was done at the University of California, San Diego in 1976 [2] and by Noll [21]. For a comprehensive reference of haptic projects see the bibliography provided in Margaret Minsky's thesis [18].

1.2 Personal HAptic iNTerface Mechanism (PHANToM)

The PHANToM is a convenient desk-top device which allows users to reach beyond the "Looking-Glass" of existing computer monitors, and actually **touch** virtual objects represented within the computer. Users connect to the mechanism by simply inserting their finger into a thimble. The PHANToM tracks the position of the user's finger tip and can actively exert an external force on the finger, creating compelling illusions of interactions with solid physical objects. Smooth spheres, flat walls, sharp corners, compliant surfaces, friction, and even texture can be convincingly conveyed to the human haptic system.

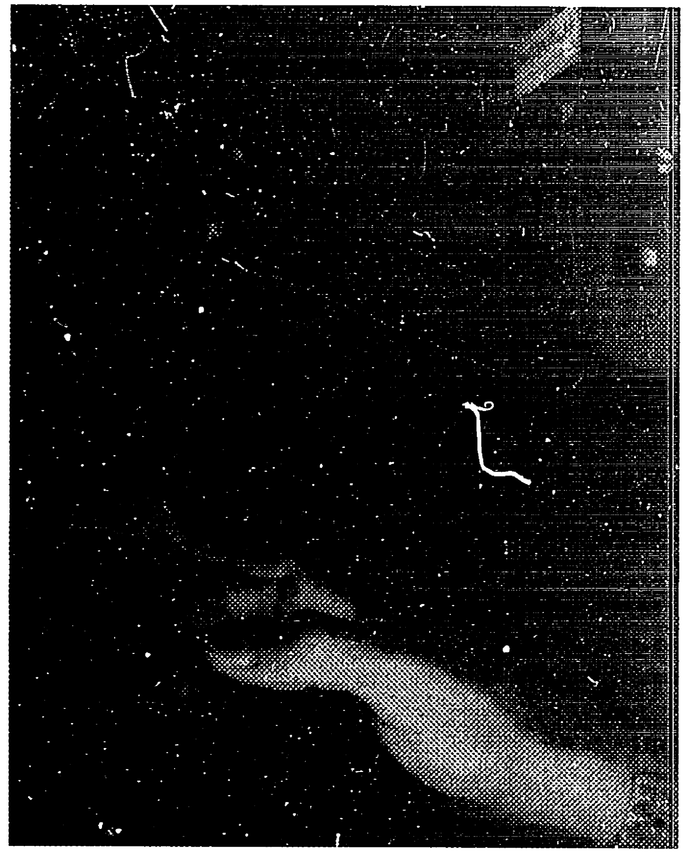


Figure 1: PHANToM

The mechanical design of the PHANToM was the topic of my bachelor's thesis [16], whereas the application and programming of the device is the primary topic of this thesis. Together the bachelor's thesis and the subsequent research described in this thesis attempt to answer the following question: *"What elements of touch can be accurately represented within the computer?"*

1.2.1 Three Enabling Observations

Three observations influenced the basic mechanical design of the PHANToM, and thereby framed the subsequent software design. The first observation established the type of haptic stimulation that the device would provide, the second determined the number of actuators that the device would require and the third established the volume or work-space that the device would possess.

1) **Force and motion are the most important haptic cues.** Kinesthetic, force, and cutaneous senses combined with motor capabilities permit us to probe, perceive and rearrange objects in the physical world. Even without detailed cutaneous information (as with a gloved hand or tool), the forces and motions imparted on/by our limbs and fingers contribute significant information about the spatial map of our environment. Information about how an object moves in response to applied force and the forces which arise when we attempt to move objects can provide cues to geometry (shape, locality, identity), attributes (constraint, impedance, friction, texture, etc.) and events (constraint change, contact, slip) in the environment. Unlike other sensory modalities, haptic interactions permit two-way interaction via work exchange. Controlled work can be performed on dynamic objects in the environment and modulated to accomplish tasks.

2) **Many meaningful haptic interactions involve little torque.** Perhaps the most significant design feature of the PHANToM is the passive, 3 degree-of-freedom "thimble-gimbal", shown in Figure 3. The decision to use the thimble-gimbal was based on the observation that many finger tip interactions with the environment involve little or no torque about the finger tip. (Tightening a screw with one's fingernail is one of the few clear counter-examples.) Because the three rotations about the center of the finger tip are neither measured nor actuated by the PHANToM, the user's finger tip can be modeled as a point or frictionless sphere in the virtual environment. The same argument applies for the tip of a stylus - the tip of a sharp pencil or pen touching a surface has virtually no torque exerted on it by the surface. Introducing three passive freedoms with the "thimble-gimbal" greatly simplifies programming as well as mechanism design.

3) **A small wrist-centered workspace is sufficient.** Many meaningful haptic interactions occur within the volume that the finger tip spans when the forearm is allowed only limited movement. In order to determine the most suitable workspace for a haptic interface, a wooden mock-up consisting of a 3 degree-of-freedom kinematic chain with a 3 degree-of-freedom thimble-gimbal was constructed. Through experience with the mock-up, it was decided that the PHANToM should be constructed such that a user could move the wrist freely without encountering the edges of the workspace. The size of a mouse pad, a sheet of note-book paper and the computer keyboard are common examples of this scale of haptic workspace.

1.2.2 Three Necessary Criteria For an Effective Interface

The following three criteria are necessary for an effective force-reflecting haptic interface device. Independent psychophysical testing could establish specifications for each of the three criteria, however available actuator, sensor, material and computer technology will ultimately determine the degree to which each of the criteria can be met. Furthermore, the three criteria must be considered simultaneously, as improving the specification for one will necessarily degrade the specifications for the other two. The PHANToM represents an effort to balance these three criteria to achieve an effective, affordable, force-reflecting haptic interface with existing technologies.

1) **Free space must feel free.** Users must not be encumbered by the device. That is, the device should exert no external forces on a user moving through free virtual space. Translated into engineering requirements, this means that there should be little back-drive friction, low inertia at the human-machine interface and no unbalanced weight. For the PHANToM, we arrived at values for each of these attributes that were perceivable, yet not distracting. Static back drive friction for the PHANToM is less than 0.1 Newton (Nt), inertia is such that the user perceives no more than 100 grams

of mass at the interface and unbalanced weight is less than .2 Nt for all points within the workspace.

2) **Solid virtual objects must feel stiff.** One metric of a force-reflecting interface is the maximum stiffness of the virtual surfaces that it is capable of representing. Because no structure or control loop is perfectly stiff, each virtual object represented through the interface must have some associated compliance. For the PHANToM the virtual object compliance is not limited by the stiffness of the structure, but rather by the stiffness of stable control that can be achieved. Using the current control algorithm, the PHANToM can reflect a maximum stiffness of about 35 Nt/cm. We have found that most users can be convinced that a virtual surface with a stiffness of at least 20 Nt/cm represents a solid, immovable wall. The maximum obtainable stiffness depends not only on the natural frequencies of the device but also on the resolution of the sensors and actuators and the servo rate.

3) **Virtual constraints must not be easily saturated.** There is nothing as disturbing as leaning against a wall and falling through it. In the virtual world, walls should be solid. The maximum exertable force for the human finger is on the order of 40 Nt [27], but during precise manipulation we find that people rarely exert more than 10 Nt of force, the peak maximum for the PHANToM. In fact, the time average force exerted during normal operation is on the order of 1 Newton, while the maximum continuous force capability for the PHANToM is about 1.5 Nt.

1.2.3 PHANToM Mechanics

In its simplest form, the PHANToM can be thought of as a transmission between three DC brushed motors with encoders and the human finger tip. The x, y and z coordinates of the user's finger tip are tracked with the encoders, and the motors control the x, y and z forces exerted upon the user. Torques from the motors are transmitted through pre-tensioned cable reductions to a stiff, lightweight aluminum linkage. At the end of this linkage is a passive, three degree of freedom gimbal attached to a thimble, Figure 3. Because the three passive rotational axes of the gimbal coincide at a point, there can be no torque about that point, only a pure force. This allows the user's finger tip to assume any comfortable orientation. More importantly, because the user can be represented by a single point or frictionless sphere within the virtual environment, collisions and resulting interaction forces with the virtual environment are easily calculated.

The PHANToM has been designed so that the transformation matrix between motor rotations and endpoint translations is nearly diagonal. Decoupling the three motors produces desirable results in terms of back-drive friction and inertia. For a haptic interface with perceivable inertia and back drive friction, it is important that the friction and inertia be nearly constant in all directions to minimize the distraction they create for the user (i.e. well conditioned inertia matrix and small, non-disparate friction components) [30].

2. Implications of a Point Force Interface

2.1 Introduction to Point Force

When the problem of building a haptic interface (the PHANToM) was presented to me in the fall of 1992, it was framed as a device which, when built, would allow for point-force interactions with a virtual environment. The device would have a three degree of freedom linkage from ground to an endpoint. (like a miniature robot arm) That part was straight forward, as others had done this before [1] [2]. But for the user connection element, Ken Salisbury suggested using a thimble in conjunction with a gimbal. The gimbal would have three passive degrees of freedom, whose axes of rotation would intersect at a single point. The passive rotational degrees of freedom meant that there could be no torque about that single point, thereby ensuring a pure force vector at that point.

Tactile interfaces which stimulate the individual receptors in the human skin have been built before [14]. Our approach of connecting to the user through a thimble meant that the PHANToM would not be stimulating the receptors in the human skin individually. Rather, pressure would be distributed over the surface of the skin, with the aggregate result consisting of a force vector that users would mainly perceive through the strain in their finger muscles.

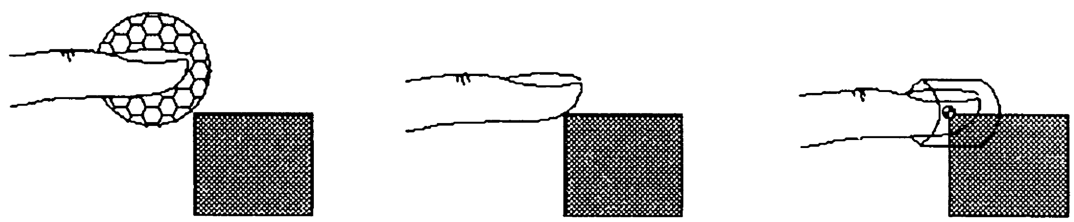
At the time the PHANToM was constructed, no one fully realized what the implications of the point force approach would be. Since then, several significant observations have been made which affect software approaches and user perceptions. These implications will be outlined in this chapter.

2.1.1 Initial Tests With The Finger Sphere

To test the concept of a force at the finger tip and the absence of individual receptor stimulation, I drilled a finger sized hole in a ping-pong ball, filled it with wax, and fitted a "finger-sphere" to my finger. My approach was not to have 20 subjects wear similar finger spheres, perform tasks, and then "objectively" evaluate the attenuated performance in certain tasks. Rather, I wanted to test the assumption subjectively for myself, and then get on to building the device.

Wearing the finger-sphere, I palpated pop-cans, shampoo bottles, sponges, and table corners. I was encouraged that I could perceive shapes, compliance, curves, attenuated corners, and to some extent, texture while wearing the finger-sphere. However, my perception of shapes was distorted somewhat by the radius of the finger sphere. For instance, a sharp corner was be perceived as having a radius equal to that of the finger-sphere (See Figure 2A). This spherical mapping would later be exploited in programming algorithms.

During unencumbered manipulation (See Figure 2B), cutaneous receptors in the skin give clues as to the sharpness of corners, even though the bulk of the finger is constrained to trace a curved path around the corner. Using a point force approach with the PHANToM's thimble gimbal (See Figure 2C) allows the user's finger to trace a very sharp path around the perimeter of a virtual corner. The fact that the user can trace a sharper, more exact path around corners using a thimble-gimbal in conjunction with the virtual environment compensates for the loss of tactile stimulation that normally aides in corner detection.



A. Wearing Finger Sphere B. Unencumbered Manipulation C. Using the Thimble-Gimbal

Figure 2: Manipulation with the Finger Tip

2.1.2 PHANToM Thimble-Gimbal

To implement the point force interface, three bearing pairs were arranged perpendicular to each other in the form of a gimbal. Because there are bearings to allow for free rotation in the roll, pitch, and yaw directions, there can be no torque about these orthogonal axes. When force is transmitted through the PHANToM to the gimbal, the force is effectively concentrated at the point where the axes of rotation coincide. This point was chosen to be inside the user's finger, so that manipulation would remain intuitive. An alternate placement for the force concentration point would have been at the surface of the skin or at the tip of the fingernail.

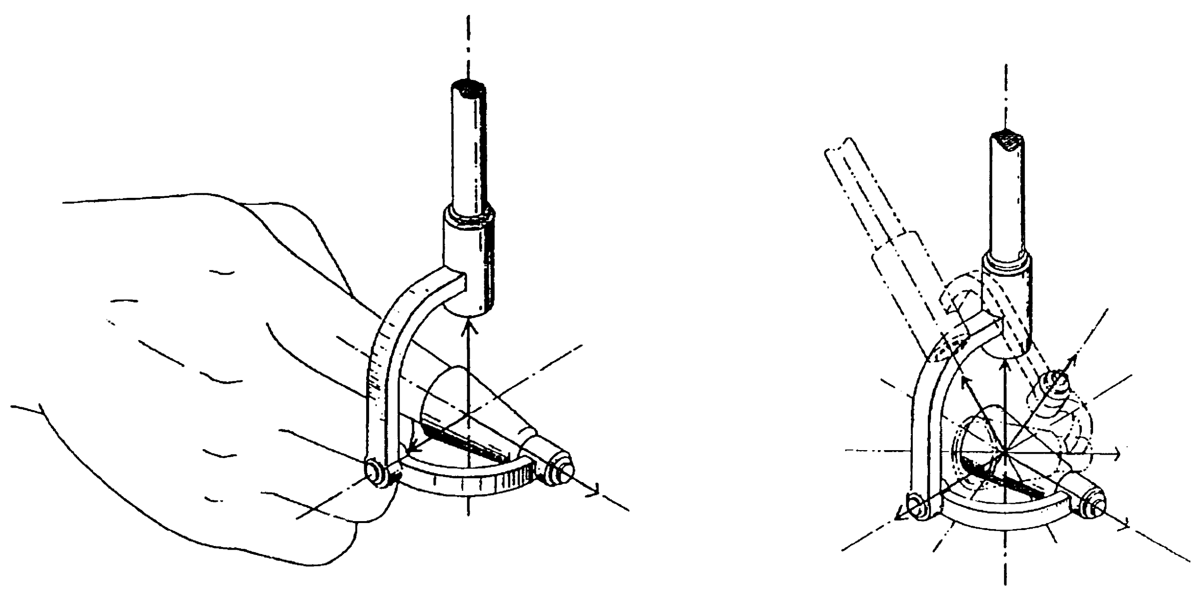


Figure 3: The Thimble-Gimbal

2.2 Increases Haptic Resoution

In one sense, representing the finger tip as a point within the virtual environment effectively increases haptic resolution beyond that of a finger tip in the real world. Cutaneous receptors aside, when a person manipulates and probes in the real world the effective resolution limited by the finger volume. However, using the PHANToM, the width of a user's finger tip in the virtual world can be made as small as data quantization permits. Manipulation remains intuitive because the thimble-gimbal allows the small virtual point to be spatially located within the user's finger tip.

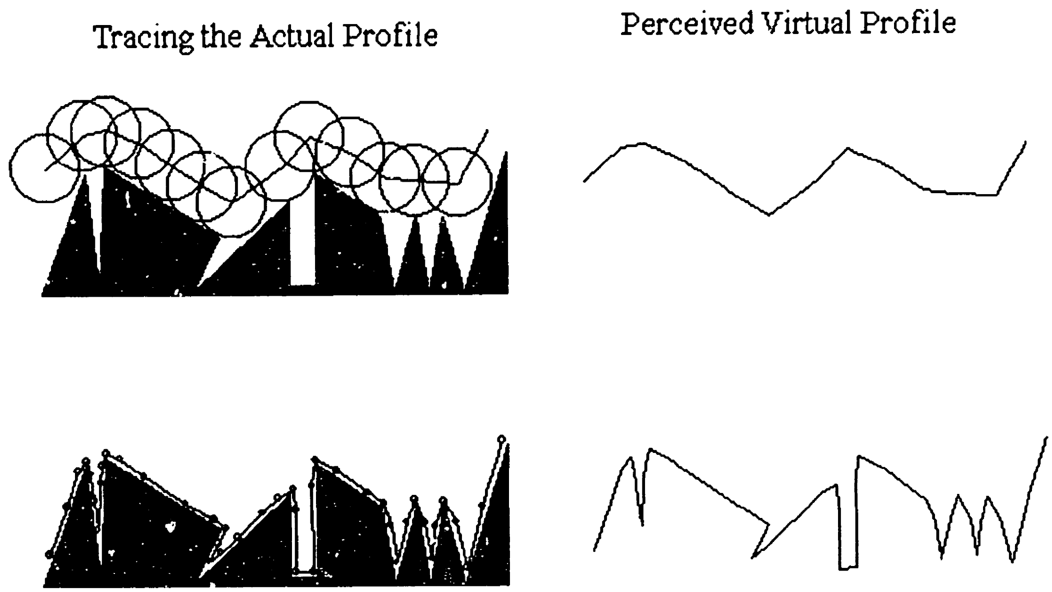


Figure 4: Radius of Manipulation Sphere Affects Haptic Resolution

2.3 Expanding the Point to a Frictionless Sphere

It is sometimes desirable to expand the virtual point to a virtual sphere. As depicted in Figure 4, using a virtual sphere filters the user's perception of the virtual model by hiding details. It might be desirable to use a virtual sphere to hide defects in the software (e.g.

floating-point round-off errors) (Figure 5), to simplify the virtual model being represented [29] (Figure 6), to represent the actual width of the user's finger (Figure 7), or to smooth sharp virtual corners (Figure 8). As shown in Figure 8, external corners can be smoothed by the expansion of the point to a sphere, but internal corners retain a discrete (sharp) transition point.

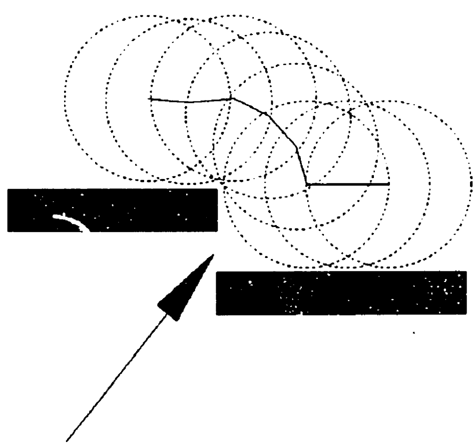


Figure 5: Hiding Program Defects

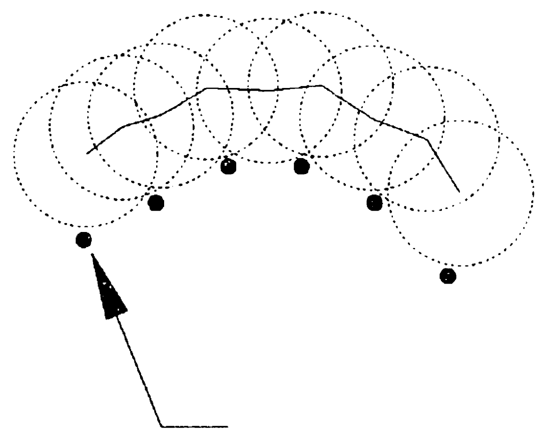


Figure 6: Programming Simplification

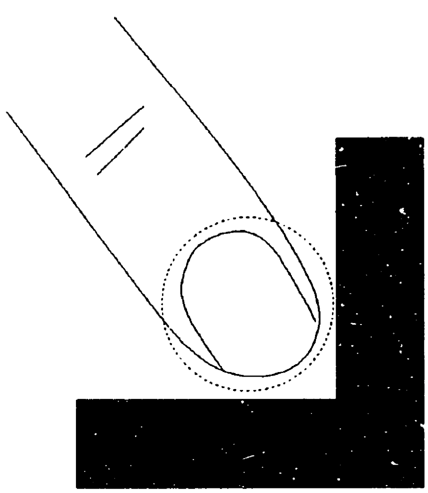


Figure 7: Representing Finger by a Sphere

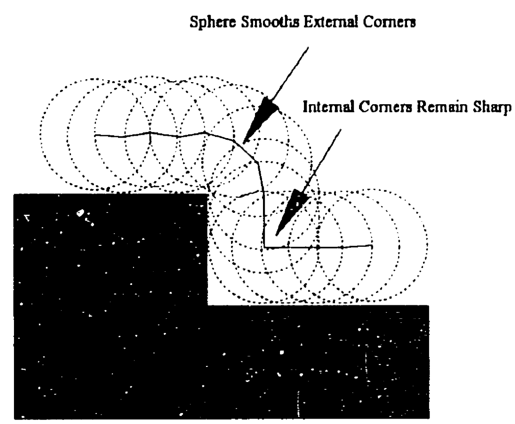


Figure 8: Smoothing Corners

It should be noted that one implication of a point force haptic interface is that it is not possible to recreate the exact friction that the sphere would encounter in the physical world. If the point force is being applied at the center of the virtual sphere, friction tangent to the sphere's surface can not be represented with a point force mechanism like the PHANToM. One component of the friction created at the sphere's surface would be a torque about the center of the sphere. A point force generator, by definition, can not generate torques. It is possible to offset the friction force from the circumference of the sphere t the center of the sphere, thereby generating an approximation of the friction force. This offset is unnoticeable for small very small spheres (3 mm diameter).

2.4 Haptic Point Force Spectrum

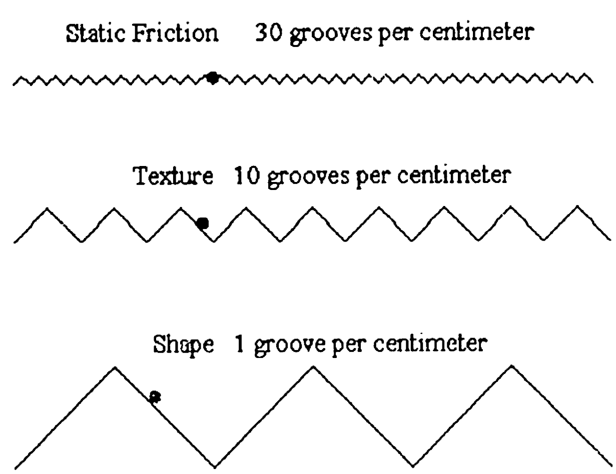


Figure 9: Haptic Spectrum of Varying Frequencies and Amplitudes

A "haptic spectrum" becomes evident when scaling virtual point force models. That is to say, the same software used to create virtual shapes can be used to generate virtual textures and even virtual static friction. The continuum of scaled surfaces is only delineated into "friction," "texture," and "shape" by human perception, so it is expected that each user will have different frequency and amplitude thresholds for interpreting the difference between "shape", "texture," and "friction." Each of the three sensations is denoted in the figure 9, with an example of the frequency required to create this perception for one user. Finding the range of frequency thresholds for different users would make an interesting topic for further study.

2.5 Feeling Through an Object

Initially, there was some concern as to how a user would adapt to using his or her finger tip to manipulate a single point in virtual space. The action is quite intuitive, with one exception. When feeling the outside of virtual spheres, some users are disturbed by the "phantom effect". That is, when using the device, one's hand can physically pass through the volume occupied by a virtual sphere, while only the finger tip is constrained to remain outside of the virtual sphere. Some users are quick to use this phenomena to their advantage and begin probing all sides of virtual objects, unconstrained by the volume of their hands. Often, first time users will reach out with a second hand to feel the virtual object, only to find it cannot be perceived with their other hand. This is a common behavior among children, adults with their eyes closed, and the visually impaired.

When the stylus is substituted for the thimble, the point force model sometimes breaks down. Imagine simulating a dentist's tool in the virtual environment. A dentist wielding the PHANToM stylus would be able to feel the tip of the tool contact the surface of the tooth. However, there would be no straight forward way to convey to the dentist the constraints of the virtual patient's mouth if the mouth were to come in contact with the side of the dentists tool. Only those forces at the tip of the tool can be accurately simulated with point force interaction.

2.6 Software Simplification

Introducing three passive freedoms with the "thimble-gimbal" greatly simplifies programming as well as mechanism design. Consider the software that would be required to interface an exoskeletal glove mechanism with a virtual environment. The program would have to detect collisions between a multiple degree of freedom virtual hand and the virtual environment, calculate reaction forces between the virtual hand and the virtual environment, and then transform these virtual forces into torque commands for the joints of the exoskeletal glove. All of these calculations must be completed to give the user wearing the glove the perception of realistic interactions with the virtual environment.

In contrast, point force interactions are simple to calculate. The algorithms must only detect collisions between a point and the virtual environment, calculate a reaction force vector, and send this force vector directly to the point (inside the thimble) being manipulated by the user. The required software is outlined in more detail in the next chapter.

3. Software Rendering Approaches

3.1 Introduction to the Basic Algorithm for Haptic Rendering

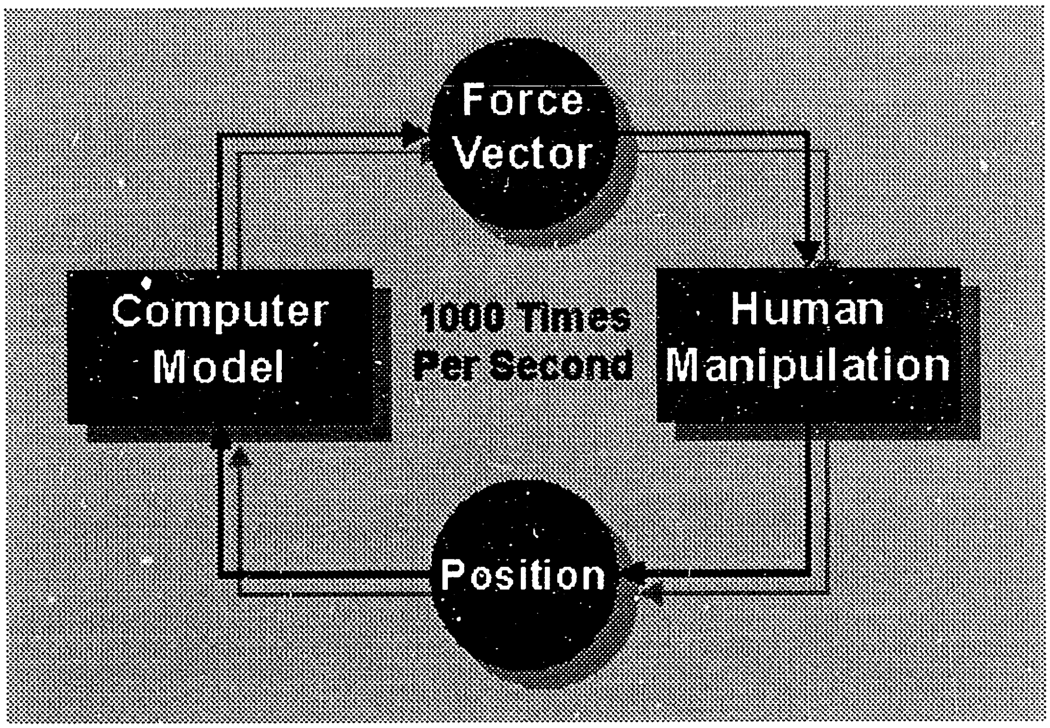


Figure 10: The Typical Haptic Software Loop

The software required to control haptic interfaces can be described as a combination of servo control algorithms and real time computer graphic rendering software. A high level description of the inner software loop would resemble the following:

A. Locate the user's finger position with respect to the virtual environment.
B. Detect collisions between the user's finger and the geometry of virtual objects.
C. Calculate a reaction force vector based on physical laws of the virtual environment.
D. Apply that force vector to the user's finger.
E. Go to A.

The most difficult steps outlined above are B and C. Collision detection is fairly straight forward, but requires clever algorithms (bounding boxes, binary space partitioning trees, quad-trees, etc.) for speed improvements in complex virtual environments. In general, for step C, a Hooke's law relationship ($f = k * x$) is used to calculate the reaction vector. That is, the reaction force is proportional to the penetration depth of the user into the virtual object and normal to the surface of the object. If the forces are always normal to the surface, the objects will be frictionless, so methods of implementing tangential forces will be outlined later (section 3.5.1).

There are several parallels between haptic rendering and real-time graphics rendering. As with computer graphic rendering, haptic rendering requires calculating surface normals across the geometry of an object. In fact typical computer graphic algorithms, such as Phong shading, can be used for haptic rendering. Just as Phong shading has the effect of smoothing the visual appearance of a faceted object by interpolating surface normals between vertices, Phong shading applied to haptic geometries has the effect of making a faceted virtual object feel smoother and therefore more realistic.

A few basic differences do exist between computer graphics and haptics software however. For instance conventional wisdom is that 30 to 60 Hertz is sufficient for "real-time" virtual environments. One must realize that this rate is only sufficient because our eyes cannot detect motion any quicker than this. As it turns out, our hands are quite sensitive to vibrations even at 200 to 300 Hertz [5]. To create convincing sensations of touch, we find that the loop must occur at an extremely high rate -- typically 1000 Hz or greater. As they say, "the hand is quicker than the eye."

Although the required update rate for haptics is about 30 times higher than that required for computer graphics, the number of required calculations for each update are much less. Consider that a high resolution computer display has about 1000 by 1000, or 1 million pixels. Each of these pixels must be updated each frame. However, a point force haptic interface has the equivalent of 1 pixel to render each frame. Because of this simplification, even standard personal computers have enough speed to execute the required haptic calculations.

3.2 Explicit Surface Definitions

The simplest geometries to model for haptic interaction are those which have explicit mathematical definitions. Because of their mathematical simplicity, the plane and the sphere were the first shapes that we rendered for interaction with the PHANToM.

In general, two calculations must occur for all geometries to be rendered hapticly. First, the software must check for collision between the user's finger tip and the virtual object. If the user is not in contact with the virtual object, then no further calculations for that particular object are required. However if a collision has occurred, a reaction force must be calculated. To compute the reaction force, a second calculation must occur to determine the surface normal at the location the user is touching the object.

Collision detection and surface normal calculations are simple for planes and spheres. For a plane represented by $Ax + By + Cz - D = 0$, one merely needs to plug the x, y, and z values of the user's finger tip into the left side of the equation. If the result of $Ax+By+Cz$ is less than 0, the user has passed beneath the plane. The vector that represents the surface normal of the plane is contained in [A B C].

The calculations are just as easy for spheres. If P is the location of the user's finger tip, C is the center of a sphere, and R is the radius of the sphere, then collision detection is a simple matter of comparing $|P - C|$ to R. If $|P - C| < R$, then a collision has occurred, and the normal vector must be calculated. The normal vector is in the direction of $P - C$. The calculations are similarly straight forward for ellipsoids and quadrics.

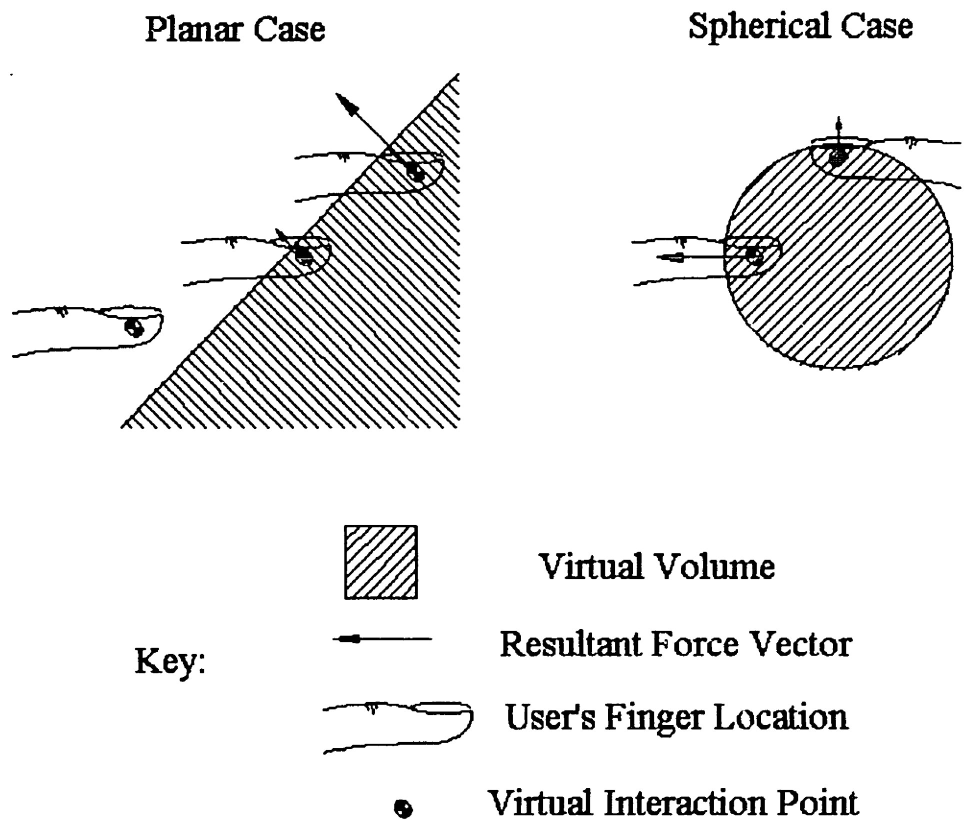


Figure 11: Resultant Force Vectors for a Plane and a Sphere

3.3 Piecewise

Complex virtual environments can be built using combinations of the basic primitives of planes and spheres described in the previous section. The simplest way of combining these primitives is to do collision detection and reaction force calculations for each instance of the primitive in the environment, and then to superimpose the reaction forces generated by these calculations. Sometimes this simplistic approach works well, but for some configurations of virtual objects, superposition breaks down. A diagram of a rectangular void (a virtual box), generated by the superposition of 4 planes is shown in Figure 12. A rectangular void is an example where superposition works well. Note that when the user is penetrating the lower left corner of the virtual box, forces from the left plane and forces from the bottom plane superimpose to give the correct corner forces.

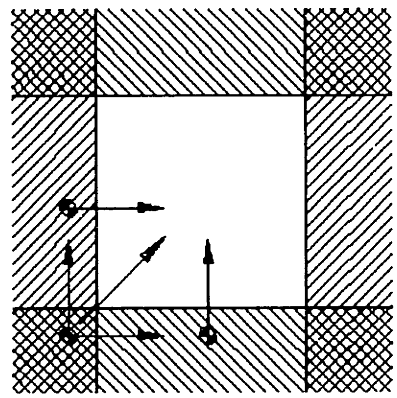


Figure 12: Superposition of Four Planes to Form a Box

In fact, superposition of planes works well for cases where the internal angle is greater than 90 degrees. However, for cases where the internal angle is less than 45 degrees, superposition begins to break down. Consider Figure 13. You will notice that in the example on the right, force vectors from the two planes tend to cancel each other out, rather than combine, thereby creating a soft pocket in the corner. Even worse, in instances of very small angles, limit cycles tend to develop as the user's finger bounces from one wall back into the other.

Superposition also works well for spheres and ellipsoids, with the same general exceptions. Where angles are greater than 90 degrees, the superposition model works well, but when the internal angle is less than 45 degrees the model begins to degenerate. The superposition of force vector for two cases is shown in Figure 14. For the case on the right, superposition does not work well. Force vectors from the two spheres tend to cancel each other out, giving the user the perception of a gap between the spheres.

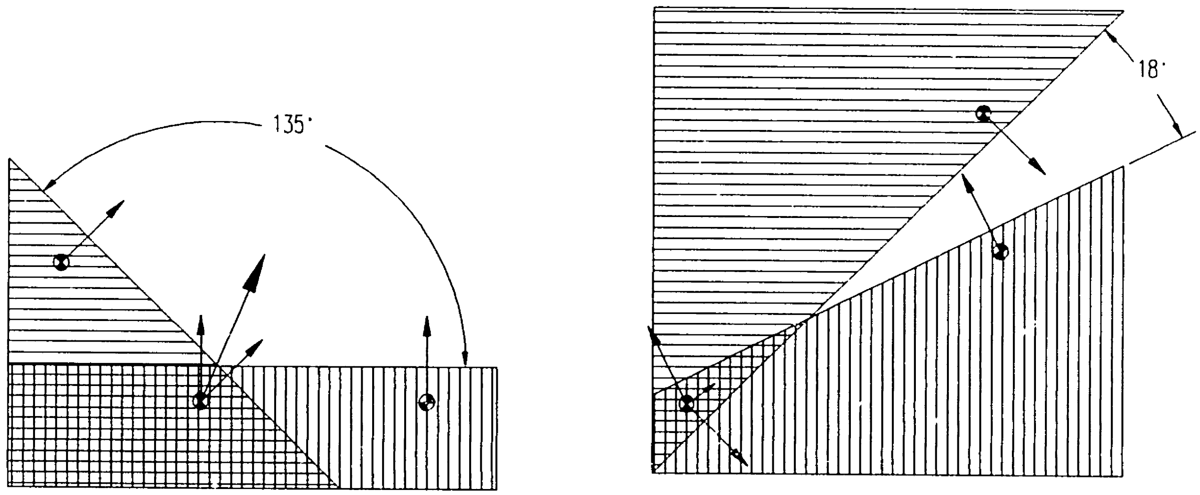


Figure 13: Comparison of Resultant Forces for Superimposed Planes Forming Acute and Obtuse Angles

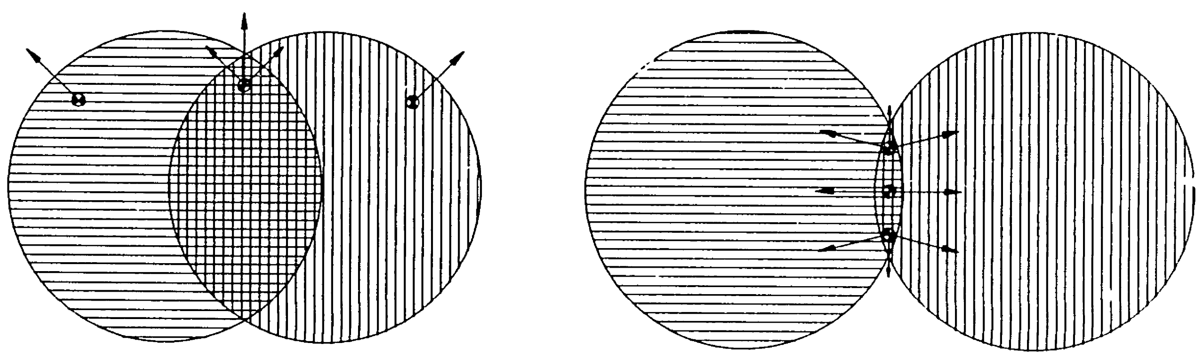


Figure 14: Comparison of Resultant Forces of Superimposed Spheres

Perhaps one of the more interesting cases in which superposition works well is for objects located within other objects. The case of a sphere beneath a plane, and the case of a sphere within a sphere are shown in Figure 15. In general, the containing object should be more compliant (have a lower stiffness, or k) than the object contained within. The feeling of a cyst beneath compliant skin or a lump within a breast can be evoked with these models. In fact, Captain Chris Hasser [11] wrote a program using the PHANToM, whereby he created the forearm of a virtual patient, complete with tough tendons and a pulsing artery, located beneath the surface of the patient's compliant skin. The entire program was written as the superposition of several instances of the ellipsoid primitive.

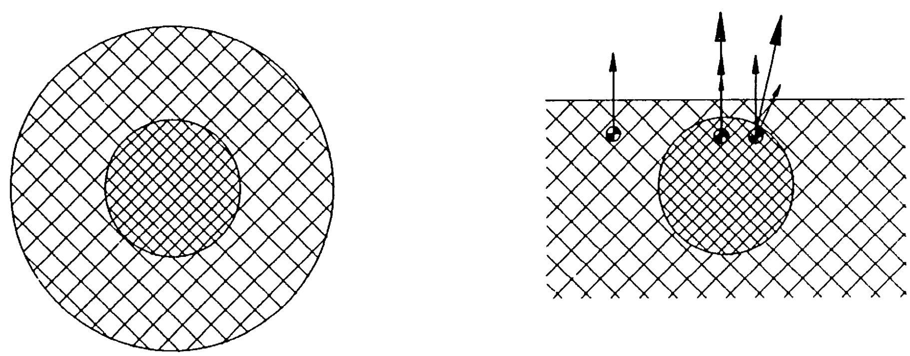


Figure 15: Spheres Superimposed with other Objects

First attempts at creating external corners with the PHANToM involved piecing together regions of planes. For cubes, this model works well. The cube is described as a volume which has a set of rules for determining reaction forces when contacted. A 2-D diagram of how the cube is partitioned is shown in Figure 16. The force vector is computed to always be normal to the nearest surface.

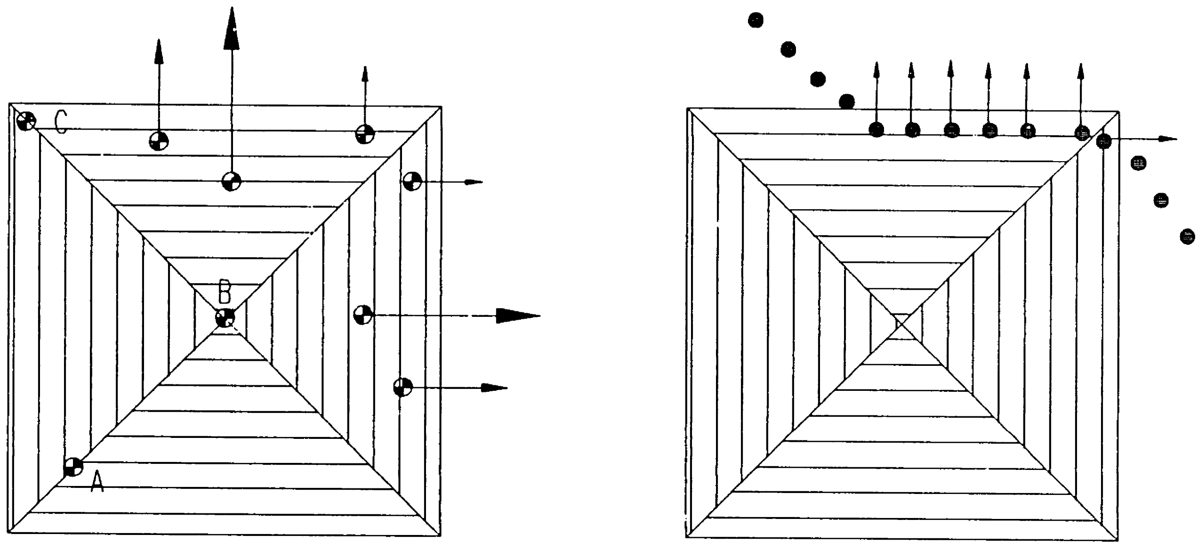


Figure 16: Piecewise Superposition to Form Cubes

For the piecewise cube model shown in Figure 16, the forces are computed based solely on the current location of the user's finger, without regard to the path the finger has taken to reach that current location. For this reason, the model is not precise. Consider the force vector that should be computed for the user positions labeled A, B, and C in Figure 16. Should the force vectors push to the right or downward? Technically, the actual force vector should depend upon the path that the user has taken to reach point A, B, or C. If the user entered the cube from the right side, the force vector should be to the right, so as to exert a restoring force which pushes the user out of the virtual volume. Ignoring the path history of the user, however, does not present a large problem for the case of cubes. In fact, this simplistic model of the cube corner led to an interesting perceptual discovery.

Consider the sensation felt by a user whose finger traces the path of the dotted line in Figure 16. As the user pushes down on and slides across the top of the cube, a restoring force is exerted upward, giving the user the sensation of a solid plane beneath his or her finger. As soon as the user because spatially closer to the right side of the cube than the top side of the cube, the direction of the force vector shifts abruptly from vertical to sideways. All of the energy that was stored as the user pushed into the top of the cube is now released as the user pops out the right side of the cube. Several users refer to the sensation as plucking a string, and most agree that the corner feels very sharp. As it turns out, users are very sensitive to direction and magnitude discontinuities in force vectors. At the corners of these cubes, a discontinuity in the direction of the force vector occurs, which aides in the perception of a corner. These discontinuities can be exaggerated to enhance corners, or smoothed (interpolated) to dull corners.

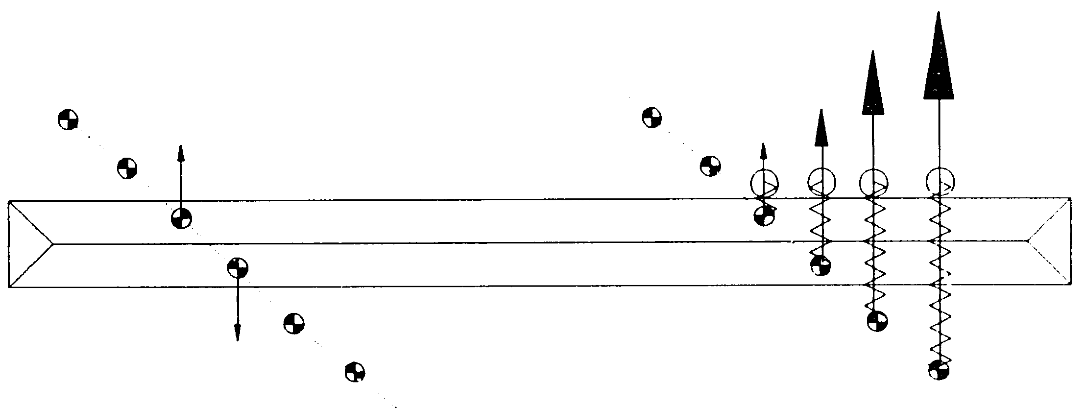


Figure 17: Resultant Forces Using No-Tracking versus Ghost Point Tracking

Although it works well for cube shaped objects, the piecewise approach to planes breaks down in many other instances when the path history of the user is ignored. Consider the long thin object shown in Figure 17. Assuming that the rule for computing forces is that the force is proportional to the penetration depth and always normal to the closest surface, observe the dotted path that a user might take. It is possible to push into the surface, realizing a slight restoring force, only to then pass through the center of the thin object and pop out the other side. For cases such as these, it is necessary for the program to track the location that a user first entered an object, and to always provide a restoring force in that general direction. One of the more compact solutions to the problem of tracking the user's past history is to track a virtual ghost point which represents the ideal location of the user's finger. The ghost point is never allowed to pass through planes and is connected to the user's actual finger position through a virtual spring and dash pot. Using a PHANToM, Zilles successfully implemented my idea for solving this problem using the "ghost" point [31] [32]. A sketch of this idea is shown on the right in Figure 17. (In Zilles' paper the ghost point is referred to as a God Object). The ghost object approach is probably the most robust and flexible approach to haptic rendering achieved to date, and has been used to model the feel of complex polyhedral objects.

3.4 Topographic Height Map

One solution to rendering arbitrarily shaped virtual surfaces is to use a height map of points. Each point in the grid represents the height of the surface at that point [8]. This solution is attractive, because the representation lends itself well to the two calculations that must occur for any haptic rendering. To reiterate, those calculations are collision detection, and surface normal calculation. A diagram of the height map representation is shown in Figure 18, and a picture of a human face stored with this representation is shown in Figure 19.

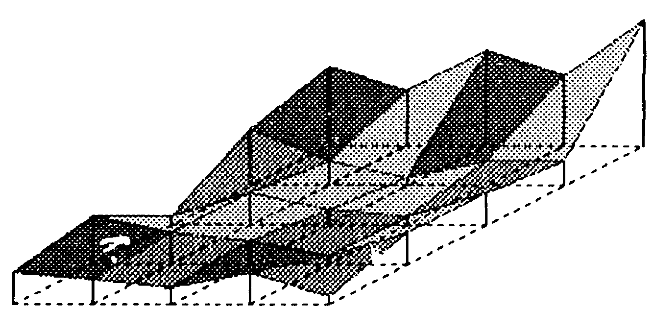


Figure 18: Height Map Representation

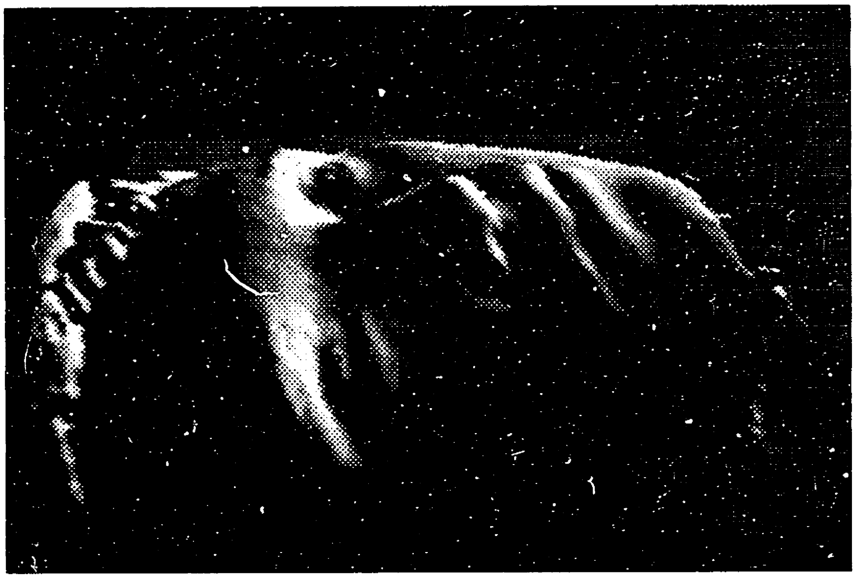


Figure 19: Human Face Represented as a Height Map

Collision detection between the PHANToM point and the surface is easy. In fact, for every X-Y coordinate, there is one unique value for Z. So, given the user's X-Y position, the program simply checks if the user's elevation (Z value) is above or below the elevation of the topography at X,Y. The elevation of the surface at any given X,Y coordinate is found by a weighted interpolation of the Z value stored at the four nearest X-Y grid points.

The surface normal is computed in the same fashion as the elevation. That is, the normal at the nearest four grip points is calculated, and the normal for the user's exact X,Y position is a weighted sum of those four grid point normals. It was with this approach that I first confirmed that Phong shading would work for haptics as well as for graphics. A 2D diagram of the surface normal interpolation is shown in Figure 20.

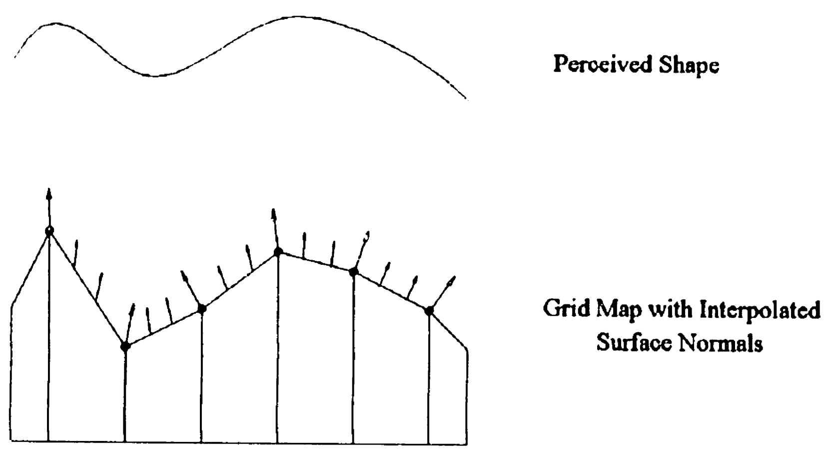


Figure 20: Surface Normal Interpolation

The biggest drawback of using a gridmap representation of virtual objects stems from the fact that each X,Y coordinate only has one Z value. Perfectly vertical walls and undercut surfaces are impossible to model with this approach, so true 3-D objects cannot be represented. This is not a problem for applications in which the user is interacting with terrain models, mathematical functions, or other models where full 3-D is not required. One interesting variation of this program was achieved by allowing grid points to be set by the user in real time using the PHANToM. In this mode, grid points were set by the user's current location, so that if the user were to trace the tip of the PHANToM stylus around a physical object, its form would be digitized within the computer. After the physical form was digitized, it could be removed from the PHANToM's workspace, and the user could then palpate the virtual representation of the object. A picture of a desk top telephone digitized with this method is shown in Figure 21.

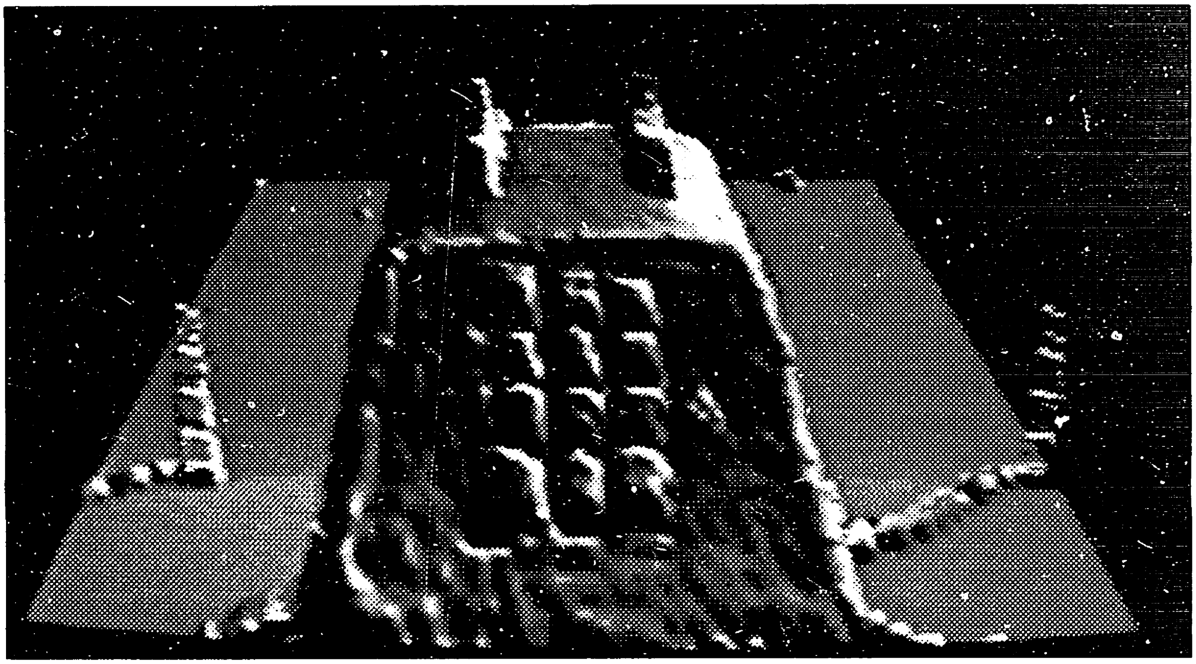


Figure 21: Height Map of Telephone Digitized with PHANToM

3.5 Surface Properties

One implication of always maintaining a reaction vector perpendicular to flat surfaces, and using $f = k*x$ for magnitude calculations is that virtual objects always feel like smooth, slippery, rubber objects. Improvements to the model are required to create the rich sensations of texture, friction, and hardness.

3.5.1 Static Friction

Although haptic sensations are all ultimately modeled with mathematical equations, it helps to have a in your mind a physical model for what is actually occurring. This is true for the case of friction. Viscous friction is relatively easy to create by introducing a force represented by the equation $f = b * v$, where b is the damping term and v is the velocity of the user's finger. However, static friction, is a bit more complicated. Static friction is that friction which keeps an object from moving, before it has begun moving.

One model for static friction is shown in Figure 22. When the user touches a surface, the point of initial contact is recorded and maintained. This point is analogous to the ghost point outlined in the previous section, but it is helpful to think of the point as a brick. If the user tries to move tangential to the plane, a spring force is exerted on the user to restore his finger back to the position of the brick. If the user pulls hard enough, he will be able to overcome the static friction between the brick and the surface, thereby displacing the brick [31] [22].

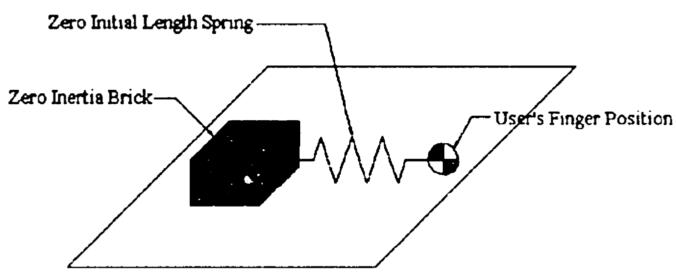


Figure 22: Physical Model of Static Friction

3.5.2 Texture

The first implementation of a texture on the PHANToM was achieved with a virtual surface defined by the equation $Z = a * \text{sine}(bX) * \text{sine}(cY)$, where a,b, and c were constants. To simplify the program a bit, instead of doing collision checking with a sinusoidal surface, the program actually did collision checking with a planar surface that moved up and down in a sinusoid, according to the user's x and y position. The difference in implementation was not noticeable to the user with one exception. Because the user was always touching a planar surface, the force vector was always vertical. A user sliding across the textured surface would feel his finger move up and down slightly in a sinusoid, but he would not feel any tangential forces. The result was that the artifical feeling of texture without friction was achieved, but texture nonetheless.

Margaret Minksy showed in her thesis that the perception of texture could be created with even a 2-D planar haptic interface [18]. The 2-D interface did not generate for forces or displacements out of the plane, but did generate forces tangent to the plane. These forces were enough to create the sensation of texture for a user.

I showed that texture could be created with only forces normal to a flat plane and Minsky showed that texture could be created with only forces tangential to the plane, but Howe, Brooks, and others have show that in general, high frequency human haptic sensors are not sensitive to direction [5] [15]. Howe successfully increased the bandwidth of a haptic interface by adding a simple speaker coil near the user connection element, thereby superimposing non-directional high frequency information on to the directinal, lower frequency force feedback.

Because the PHANToM is a point force interface, it is possible to model the complete virtual environment, calculate reaction forces, and then as the last step map a spatial force disturbance onto the force vector to be sent to the PHANToM point. Morgenbesser and others have used this form of haptic "bump mapping" to created texture in this way [19].

3.5.3 Compliance and Hardness

No haptic interface can be made perfectly stiff. Mechanical limitations aside, no motor controller can have an infinite gain. Therefore, virtual objects will always have some compliance. Even though solid objects within the virtual environment are slightly compliant, users are often willing to accept that the objects are solid. Perhaps users tolerate this amount of compliance because it is on the order of the compliance of the human finger pad. Also, the fact that users can effortlessly slide tangent to the walls seems to reinforce the illusion of a solid surface. A stiffness of 20Nt/cm seems sufficient to convince users that their fingers are touching solid objects.

Stiffness is not the only metric required to create very hard virtual walls. Consider a wall that is modelled by a lossless spring that is infinitely stiff. If you were to punch this wall, your fist would return with the same velocity it had when it struck the wall, and in fact, you would not perceive the wall as hard. A wall's coefficient of restitution contributes to its sense of rigidity, so damping is required to complete the model of a hard wall. If a dash pot in parallel with a spring is used to model a stiff wall, then the wall will feel harder than if it were merely modelled with a spring. Not only does the dash pot dissipate energy when the user strikes it, but it also creates an step function of force when the user moves from a region of freespace (where f=0) to the region of the wall where the dash pot adds the force component (f=b * velocity, and velocity is non-zero).

Going even one step further, researchers have modelled the impulse required to exactly cancel the inertia of the user travelling through freespace. When the user strikes the virtual wall, in addition to the dash pot and spring model, an impulse of energy is applied which will theoretically exactly cancel the inertia of the user. Because convincingly hard walls is one of the most important criteria for an effective haptic interface, much research has been done in this area [6] [22].

While a user tapped a virtual floor with the PHANToM, recordings of displacement versus time were taken for both the spring model and the spring + dash pot model. These plots are shown in Figure 23. Note that for the model of a spring + dash pot, forward movement of the user into the wall was stopped sooner, and oscillations died out quicker than for the simple spring model.

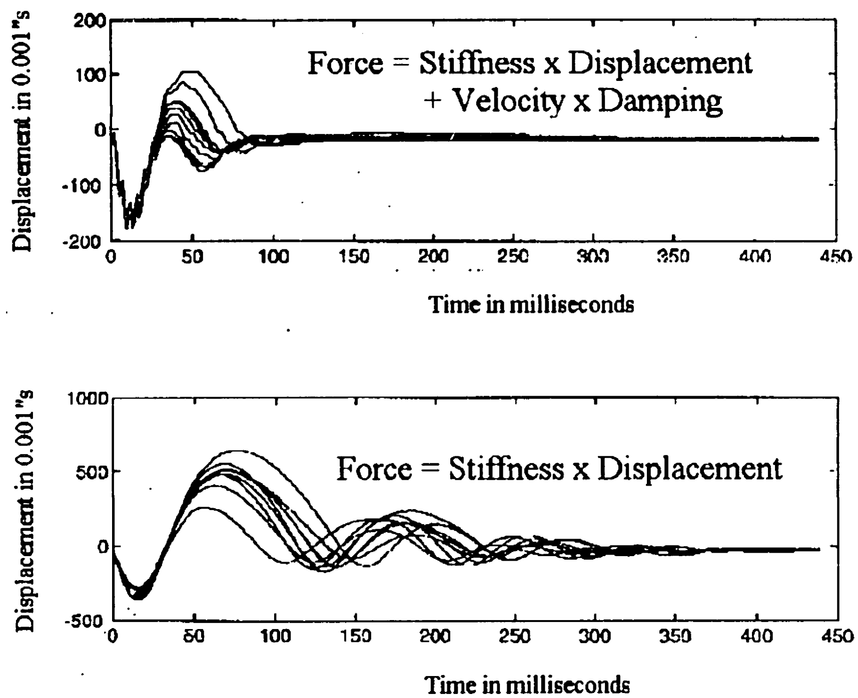


Figure 23: A User Tapping Damped and Undamped Walls

3.6 Buttons and Switches

The button is a metaphor that has already gained wide use in graphical user interfaces. With haptic interfaces, it is now possible to turn the metaphorical buttons into literal buttons which can be touched, pushed, and activated. Furthermore, these buttons provide tactile feedback when they click. In fact, virtual buttons can be assigned various stiffnesses to convey purpose. In the demo portrayed in Figure 24, each of the buttons has a different stiffness, ranging from a light keyboard touch to a heavy circuit breaker feel.

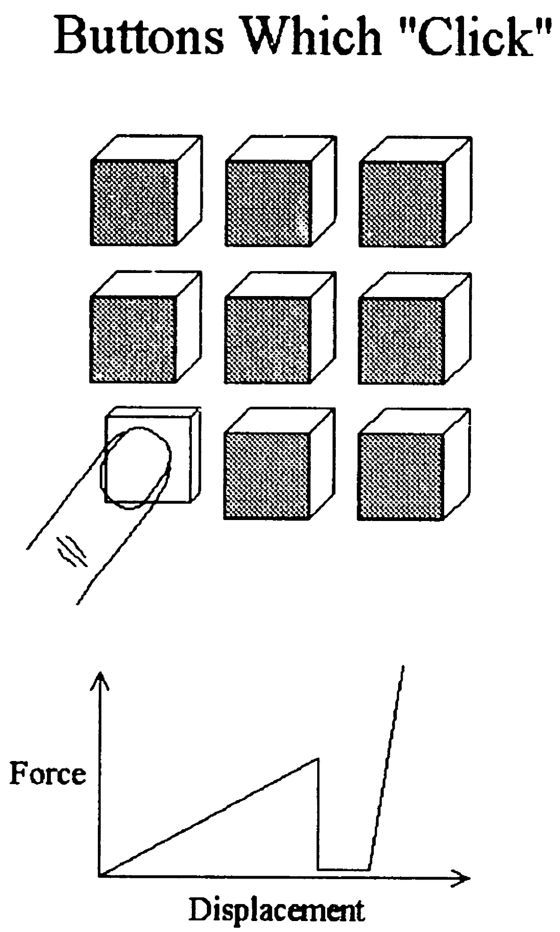


Figure 24: Virtual Buttons and a Typical Force-Displacement Curve

There are two slopes in the force-displacement graph in Figure 24. The lesser slope, beginning at the origin represents the restoring spring within the button. The steeper slope represents the wall behind the button that the user encounters after depressing the button fully. The force-displacement graph is discontiuous at the end of the first slope. This discontinuity causes the sensation of a "click" when the user depresses the button. Even when this discontinuity is small, users are able to perceive it.

Another demonstration of virtual switches is portrayed in Figure 25. In this demostration, cubes are constrained to slide only back and forth on a plane. In addition to the properties of mass, viscosity, stiction, and surface stiffness, each cube has an underlying characteristic spring function.

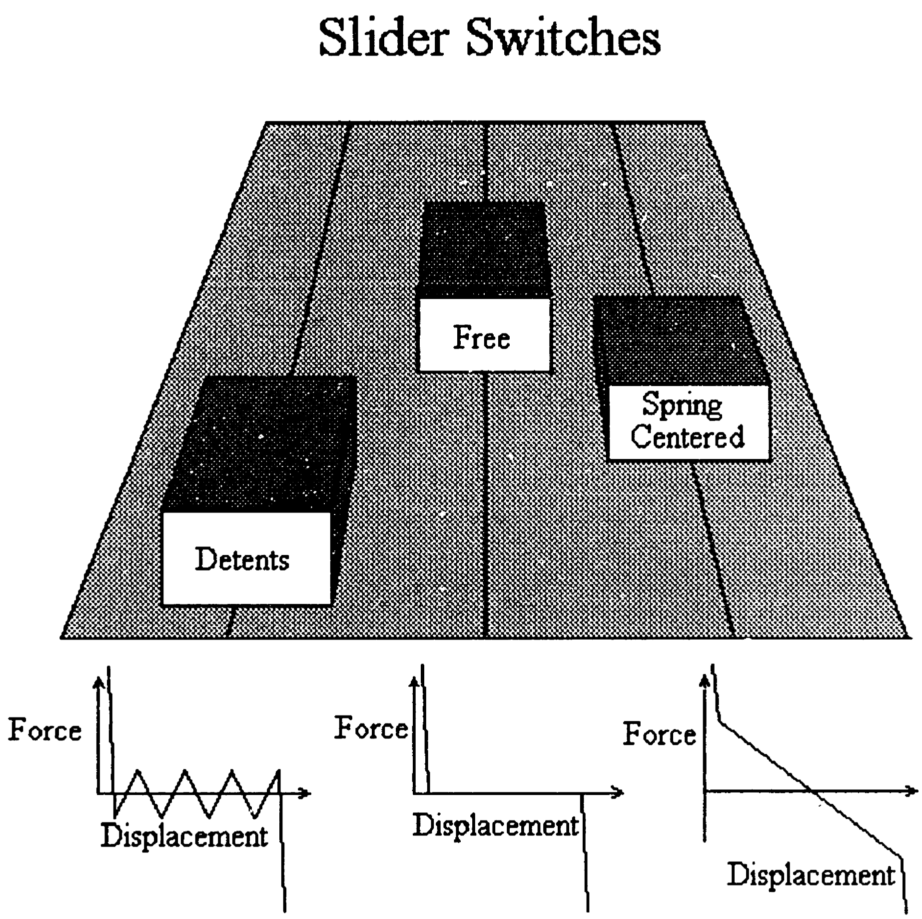


Figure 25: Sliding Switches and Their Associated Force-Displacement Curves

Beyond the obvious need for them as metaphors for graphical user interfaces, virtual buttons and switches are useful for virtual prototyping applications. Engineers could make good use of these haptic primitives to design interfaces to VCRs, computer keyboards, and car dash boards.

3.7 Virtual Clay

Clay is a very interactive and engaging medium in which the hands are used to shape forms. A virtual clay programs was written for this thesis to test the validity of using haptic interfaces to interact with virtual clay.

3.7.1 Present Implementation

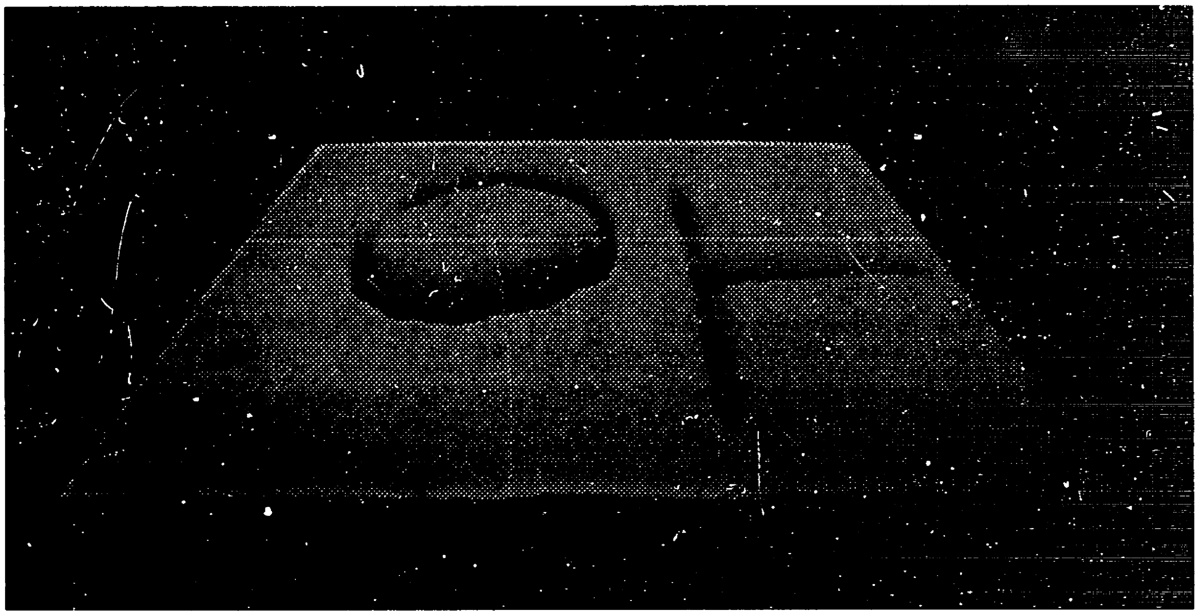


Figure 26: Carvings in Virtual Clay

The current implementation of virtual clay is a variation on the height map program described in section 3.4. Just as with the height map program, the user can feel a miniature virtual terrain beneath his fingers. However, if the user exerts force on the surface beyond a preset threshold, the surface will begin to deform. If the user exerts more pressure, the terrain defo[illegible]cker. The terrain deformation is achieved computationally by lowering the 4 nearest grid points, and then by lowering the 8 next nearest grid points by half as much. As a result, if the user were to push down continuously, a parabolic shaped hole would form in the surface of the clay.

Although the implementation is very simplistic, a very promising discovery was made in the course of testing the program. Users found that cutting straight and circular grooves in the surface was very intuitive. In fact, hemispheres could be cut from a flat surface by very carefully tracing a circle while exerting some pressure on the surface. After making the initial circular groove, completing the hemisphere was trivial and could be done quickly even with no visual feedback. As the user's finger traced the groove to complete each deeper cut, it was found that the groove left from the previous cut guided each subsequent cut. This is a natural behavior with real clay, and could not be duplicated without haptic feedback.

3.7.2 Future Implementations

Future implementations of virtual clay could be significantly enhanced by adding several realistic features. First would be an algorithm for conservation of volume. Real clay does not compress or disappear when you exert force on it, but the present virtual clay does.

A menu of cutting tools would allow designers to develop different techniques for forming the clay. Using the present cutting tool is similar to molding the clay with a ping-pong ball. Another nice feature would be to change the behavior of the cutting tool according to its angle with repect to the surface of the clay.

It would be desirable to be able to mold the clay into true three dimensional forms. Presently, the user can only push down from the top or sides of the clay. There is no way to form under cuts. Perhaps a polygonal approach with the ghost object described in section 3.4 or a volume rendering approach using voxels would work nicely.

Finally, the ability to manipulate the position and orientation of clay with left hand while molding with the right hand seems very important. Positioning the clay with the left hand could probably be done with a simple position tracker, as it is not clear to me whether the force interaction between the right and left hand is important when the left hand is holding a heavy object. (If you are left handed, please exchange the words "left" and "right" in the preceding paragraph.)

4. Applications

The applications for haptic interfaces are almost as numerous as the applications for personal computers and workstations. However, until they gain wide acceptance, the price of haptic interfaces dictates that they will be used in critical applications where they can justify their cost several times over. Some of the more promising applications are described briefly in this chapter. Each of these areas lends itself well to the point force approach to haptics, therefore PHANToM users are already writing the code to enable these applications.

4.1 Computer Interface for the Blind

It is ironic that graphical user interfaces, mice, windows, and icons -- innovations that have made it easier for most people to use computers -- have made it more difficult for the blind to use computers. In general, the visually impaired use text to interact with computers. Whether they use speech synthesizers, touch typing, or braille devices, the blind are often interacting with computers temporally, not spatially. Using a touch interface like the PHANToM, the blind can take advantage of the spatial nature of today's computer applications.

The blind could use the PHANToM to feel 2D and 3D mathematical surface plots produced by graphing programs, access the database of geographical maps available on CD ROMs, and even interact with windows environments by feeling buttons and by probing constraints of the window's perimeters.

Labs are already using the PHANToM to develop software applications for the blind. At Lund University in Sweden, they have developed a painting program where users can feel the art that they or others have created [28]. Each of 10 colors in the program has a distinct, recognizable texture that can be felt with the PHANToM. The same group at Lund University has developed mathematical plotting programs and a virtual version of the electronic battleship game for the blind, complete with "rough, wavy seas" that can be felt by the players.

4.2 Medical Training

For many of the same reasons that flight simulators make sense, virtual medical training simulators are a good idea. Just as with flight simulators, many scenarios can be simulated ranging from routine tasks to risky, hard to perform procedures. Identical scenarios can be recreated, operators can "undo" mistakes for the sake of training, and performance can be objectively measured and evaluated. In fact, these simulators could one day serve in the process of certification. Just as with flight simulators, there is no substitute for the real thing, but practicing on the virtual could augment and enhance conventional training.

Haptic feedback is a critical element of most medical procedures. Surgeons may operate in areas not visible to the eye, but they never operate with their hands tied behind their backs. Haptic feedback is also a critical element in diagnosis and in many nursing procedures. So it is clear that haptic interfaces have a role in virtual medical training simulators.

The PHANToM haptic interface has already enabled many medical aplications. These applications include surgery, dentistry, and needle insertion procedures.

4.3 Computer Aided Design

Millions of designers are using two dimensional input devices and two dimensional graphic displays to design three dimensional parts. There has probably been as much or more research on the methods of adapting two dimensional user interfaces to interact with three dimensional parts than there has been research on developing three dimensional user interfaces. Not only does the PHANToM allow for three dimensional input from a user, but it also allows for three dimensional feedback to the user. For this reason, it is an natural interface to CAD systems.

The PHANToM allows users to actually feel mechanical constraints of virtual parts. It will allow designers to more intuitively develop free form shapes, with enabling code like the virtual clay program. Going a step further, a PHANToM interface to a virtual prototyping program will allow designers to feel and even assign functionality to their designs. In fact, fit and assembly of mechanical parts (virtual) can be tested intuitively using the PHANToM haptic interface [10].

4.4 Interface to the Microscopic World.

Researchers from the Computer Science Department and Physics Department at the University of North Carolina, Chapel Hill have developed a virtual reality interface to scanning probe microscopes [9]. Referred to as the "nanomanipulator," the system uses a PHANToM to provide a direct link between a person and the tip of an atomic force microscope(AFM). The nanomanipulator allows users, in real time, to see and touch actual particles as small as 20 nanometers. Using the PHANToM, the operator can increase the force exerted by the tip of the AFM to modify the microscopic surface. At UNC, they have been able to bend, break, and move particles of tobacco-mosaic virus, examine fruit-fly chromosomes, and manipulate gold colloids.

5. Conclusions

Haptic interfaces will change the way the world uses computers. But why haven't they already? There are several reasons why they haven't already and these barriers are quickly disappearing.

1. **Computers weren't fast enough. Now they are.**

   Computers existed in 1985 that were fast enough to control haptic interfaces, but these weren't the computers sitting on your desk. Cheap, wide spread computing power is now available. When the PHANToM was developed in 1993, colleagues encouraged me to move it off the PC onto a higher-end multiprocessor computer but I resisted because I wanted a PHANToM to take home. Home computers have the necessary number crunching power to calculate haptic interactions at the required 1000 hertz.

2. **The raw materials didn't exist. Materials science has changed that.**

   The raw materials for building good haptic interfaces could be found 10 years ago, but only recently have exotic materials such as carbon-fiber composites and rare-earth magnets found commercial applications such that their price is reasonable and their performance reliable. Fundamentally, materials and tolerances set the cost of goods, and it is now possible today to produce reasonably priced, functional haptic interfaces in large quantities.

3. **Haptic interfaces weren't designed well. Point force is the correct approach.**

   The PHANToM is not perfect, but it avoids many of the common mistakes made in previous designs of haptic interfaces. When previous attempts at haptic interfaces failed to give convincing sensations, there was a tendency to add more degrees of freedom, increase the maximum force threshold, or go in a different direction altogether. Most of these directions were wrong. Increasing powered degrees of freedom always increases cost, decreases bandwidth, and complicates programming -- three DOF is enough. Increasing the maximum force threshold means increasing the background friction, and as we have found, the ratio of force to friction is much more important than absolute force. Inflatable air bladders, roaming exoskeletons, and buzzing vibro-tactile stimulators all now seem to be misguided attempts to achieve virtual touch.

We simply didn't understand what touch was and how humans used it to interface with the real world. Salisbury's [24] work with building and programming robot hands gave him the insight into human manipulation required to frame the haptic interface problem correctly. He realized that 3 degrees of freedom of point force at the finger tip or tool tip enables most of dextrous human manipulation. In fact, his robot hands were built with no palm and little if any facility for grasping with anything but the hemispherical finger tips. This way of thinking carried over to the PHANToM in the form of a thimble-gimbal. The thimble-gimbal is perhaps the single most important design element in the PHANToM.

4. **Software was not available. A critical mass of software developers has emerged.**

There is always a chicken and the egg problem with new hardware that requires software. No matter how well designed, the hardware will never have critical functionality without software. There are now more than 50 PHANToMs in the United States, Europe, and Japan, with software development being conducted at each site. Make no mistake, we have reached critical mass (and growing) for software development. All of the haptic interfaces will not mysteriously disappear one day and be forgotten. Rather thousands, if not millions more will be distributed to computer users in the next decade, as prices are reduced while applications are developed.

The purpose of the research in this thesis was not to develop packaged software applications or to do basic psycho-physical research. Rather, I wanted to show by demonstration that the PHANToM haptic interface is a feasible hardware technology worthy of further software development. I wanted to explore and be the first to try basic concepts, but I also wanted to inspire others to do better and go further than me. In that regard I have been moderately successful in that at least six other theses dealing with software development for the PHANToM were ironicly completed before mine! [10][12][13][19][29][31] This thesis has shown the feasibility of point force haptic interfaces. More software will be developed and designs will change, but it is clear that the point force approach to haptic interfaces will remain the predominant paradigm for touch interaction over the next several years.

APPENDIX A: Informal Haptic Dictionary

haptic illusions - software and hardware tricks used to enhance or simplify the computer models of physical sensations. These are "tricks" in that they are not based strictly on the mathematical models of the real world, yet manipulating these models elicits convincing physical sensations of touch.

haptic rendering - analogous to computer graphics rendering. It is likely that this field will see major advances in the next 3 years.

hapticly - used like visually.

hapticly challenged - those people who have difficulty sensing and manipulating with their hands. About 1 in 500 people are hapticly chalenged, and therefore receive little or know benefit from using a haptic interface to interact with computers.

haptician - one who performs haptics. Like a magician or a musician.

PHANToM™ - . Personal HAptic iNTerface Mechanism. [20]

Thimble-Gimbal™ - A thimble surrounded by a three degree of freedom passive gimbal. This enables intuitive point-force interactions at the finger tip.

Vibro-Tactile - A buzzing element which stimulates the receptors in the skin. Usually actuated with a piezo element, a shape-memory alloy, or a voice-coil speaker. Does not facilitate input from the human to the computer.

Bibliography

[1] Adelstein, B. D. "A Virtual Environment System for the Study of Human Arm Tremor," PhD Thesis, *Department of Mechanical Engineering, Massachusetts Institute of Technology*. 1989.

[2] Atkinson, William D., Karen E. Bond, Guy L. Tribble III, and Kent R. Wilson. "Computing with Feeling", *Comput. and Graphics*, Vol. 2, pp. 97-103. Pergamon Press, 1977 Great Britain.

[3] Batter, J. and Fredrick P. Brooks, Jr. "GROPE-1: A Computer Display to the Sense of Feel" *Inform. Process.* 71, 759 (1972)

[4] Brooks, Fredrick P., M. Ouh-Young, J.J. Batter, and P.J. Kilpatrick. "Project GROPE - haptic displays for scientific visualization" In Computer Graphics, Vol 24-4, pages 177-185, Dallas Texas 1990

[5] Brooks T.L., "Telerobotic Response Requirements." *Procedings of 1990 IEEE International Conference on Systems, Man, and Cybernetics.* pp. 113-120. Los Angeles, CA, November 1990

[6] Colgate, J. Edward., P. E. Grafing, M. C. Stanley, and G. Schenkel. "Implementation of Stiff Virtual Walls in Force Reflecting Interfaces." in *IEEE Virtual Reality Annual International Symposium*, 1993.

[7] Cybernet Corp., manufacturers of the PER-force 6 axis joystick. Ann Arbor, Michigan.

[8] Dench, Phil and Jill Smith, *Curtin Univerity of Technology, School of Computing*, personal communication and development of PHANToM software for height map representation. 1994

[9] Finch, Mark, Vernon Chi, Russell M. Taylor II, Mike Falvo, Sean Washburn, and Richard Superfine. "Surface Modification Tools in a Virtual Environment Interface to a Scanning Probe Microscope." *Computer Graphics: Proceedings of the ACM Symposium on Interactive 3D Graphics* (Monterey, CA, April 9-12, 1995): 13-18.

[10]Gupta, Rakesh. "Prototyping and Design for Asembly Analysis Using Multimodal Virtual Environments," PhD. Thesis, *Department of Mechanical Engineering, Massachusetts Institute of Technology*. 1995.

[11]Hasser, Christopher J. Wright Patterson Air Force Base, personal communication and development PHANToM software for pulsing wrist demonstration.

[12]Hasser, Christopher J. "Tactile Feedback for a Force-reflecting Haptic Display." M.S. Thesis. *University of Dayton*. Dayton, Ohio. December 1995.

[13]Hoeg, Hans D. "Development of Virtual Objects of Arbitrary Shape", B.S. Thesis, *Department of Mechanical Engineering, Massachusetts Institute of Technology.* 1995.

[14]Kazmarek, Kurt. A., Mitchell Tyler, Paul Bach-y-Rita "Electrotactile Haptic Display on the Finger Tips." *Annual International Conference of the IEEE Engineering in Medicine and Biology Society - Proceedings.* Baltimore, Maryland 1994.

[15]Kontarinis, Dimitrous A, J. S. Son, W. Peine, R. Howe "Tactile Shape Sensing and Display System for Teleoperated Manipulation" Proceedings of the 1995 IEEE International Conference on Robotics and Automation. Nagoya, Japan. 1995

[16]Massie, Thomas H., "Design of a Three Degree of Freedom Force-Reflecting Haptic Interface," SB Thesis, *Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.* May 1993.

[17]Massie, Thomas H. and Salisbury, J.K. "The PHANToM Haptic Interface: A Device for Probing Virtual Objects." *In ASME International Mechanical Engineering Exposition and Congress*, Chicago, IL November 1994

[18]Minsky, Margaret. "Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display." PhD Thesis, *Massachusetts Institute of Technology*, 1995.

[19]Morgenbeŗser, H. B. Force shading for shape perception in haptic virtual environments. Masters Thesis submitted to *the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology*, 1995.

[20]Nolan, Sean and Grant Eiselen, Massachusetts Institute of Technology, personal communication - helped name PHANToM. 1993

[21]Noll, A. M., "Man-machine Tactile Communication," *J. Soc. Inform. Dis.*, July-Aug (1972)

[22]Salcudean, S.E., T.D. Vlaar "On the Emulation of Stiff Walls and Static Friction with a Magnetically levitated input/output device" *In ASME International Mechanical Engineering Exposition and Congress*, (Chicago, IL November 1994)

[23]Salisbury, J.K., D. Brock, T. Massie, N. Swarup, C. Zilles "Haptic Rendering" Programming Touch Interaction with Virtual Objects" *Procedings of the 1995 Symposium on Interactive 3D Graphics*. Montery, California 1995

[24]Salisbury, J. Kenneth. and Mason, M.T. *Robot Hands and the Mechanics of Manipulation*, MIT Press, Cambridge, MA 1985

[25]Sarcos, Inc., manufacturers of the Sarcos Exoskeleton. Salt Lake City, Utah.

[26]Srinivasan, M.A. and J.K. Salisbury "Chapter 4: Haptic Interfaces," in Virtual Reality: Scientific and Technical Challenges, Eds: N. I. Durlach and A. S. Mavor. National Academy Press, Washington D.C., 1994.

[27]Sutter, P.H., J.C. Iatridis and N.V. Thakor, "Response to Reflected-Force Feedback to Fingers in Teleoperations," *Proc. of the NASA Conference on Space Telerobotics*, NASA JPL 1989.

[28]Svenson, Lars, Ulf Larson, Hakan Neveryd, and Charlotte Magnuson, Lund University, Sweden, personal communication and development of PHANToM software for the blind. 1995

[29]Swarup, Nitish. "Haptic Interaction with Deformable Objects Using Real-Time Dynamic Simulation" M.S. Thesis, *Department of Mechanical Engineering, Massachusetts Institute of Technology*. 1995.

[30]Vertut, J. and P. Coiffet, *Robot Technology, Teleoperation and Robotics: Evolution and Development, Vol 3A, and Robot Technology, Teleoperation and Robotics: Applications and Technology, Vol 3B*, Prentice Hall, Englewood Cliffs, NJ. 1986

[31]Zilles, Craig .B. "Haptic Rendering with the Toolhandle Haptic Interface," M.S. Thesis, *Department of Mechanical Engineering, Massachusetts Institute of Technology*. 1995.

[32]Zilles, Craig B., J. K. Salisbury "Constraint Based God Object Method for Haptic Display", *Procedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Pittsburgh, Pennsylvania 1995