DSC-Vol. 55-1, Dynamic Systems and Control:
Volume 1
ASME 1994

ON THE DEVELOPMENT OF A FORCE-FEEDBACK MOUSE AND ITS INTEGRATION INTO A GRAPHICAL USER INTERFACE

A.J. Kelley†
Department of Precision Machinery Engineering
University of Tokyo
Tokyo, Japan

S.E. Salcudean
Department of Electrical Engineering
University of British Columbia
Vancouver, British Columbia
Canada

ABSTRACT

A novel computer input-output device utilizing force feedback has been developed. Elements of electromechanical and controller design are discussed. Issues relating to hardware/software integration with a workstation and the X Windows graphical user-interface are presented. Results from experiments demonstrating the ability of the system to simulate linear sliders, mechanical limit stops such as the edges of the display or of a window, and mechanical push-buttons are given. Based on these results, several design guidelines and possible uses for force feedback in general purpose graphical user-interfaces are proposed.

INTRODUCTION

Despite the extensive proliferation of computer input devices for graphical user interfaces in the last decades, since the introduction of the first mouse-style input devices there have been few significant technological advances in commercially available pointing devices. As computer processing power improves, the tools which interface users to their computers can become the most significant productivity bottlenecks in the work process. While users have their choice of mice, joysticks, tablets, light pens, knob boxes, and touch screens, these devices all share a common trait beyond their "point-and-pick" functionality: they are passive tools that do not provide direct feedback to the user.

In this work, we are attempting to integrate force feedback concepts that have been developed for telemanipulation operators in the past with "point-and-pick" devices to allow graphical user interfaces to take advantage of the user's haptic senses. The fundamental idea of such an application for this technology was originally considered by Hannaford et al [9] based on work in force feedback interfaces for teleoperation, but no implementation and only a brief discussion have been provided. A variety of anecdotal applications for haptic computer interface and related telemanipulation technology have been discussed by Atkinson (1977), Batter and Brooks (1972), Davis (1990), Fisher et al (1987), Foley (1987), Hollis et al (1991), Iwata (1990), Ming Ouh-young et al (1988, 1989), Minsky et al., Noll (1972), Patrick et al (1990), Wiker et al., Williams (1989), Sheridan (1992), Stix (1991) and Sutherland (1965).

Our research is motivated on the premise that a haptic interface could contribute to a measurable increase in user performance, speed, and efficiency with lower levels of mental and physical stress. We are exploring how the graphical objects associated with a conventional window-based graphical user interface can be augmented with mechanical properties that will make it easier and more fun to use. For this purpose, we have: (i) designed and built a 2-DOF electromechanical haptic interface, called "MagicMouse", that exceeds most accepted performance figures (position and force frequency response) necessary for "transparency" in teleoperation systems; (ii) integrated the device with a workstation and the X-Windows system; and (iii) developed methods for the emulation of mechanisms associated with graphical objects (window borders, sliders, push-buttons, icon gravity sinks). Issues of hardware interface, control and communication software have been addressed. The look and feel of a user interface that makes use of the mechanical primitives experimented with are also discussed.

†Work done at the University of British Columbia, 1991

ELECTROMECHANICAL DESIGN

Many of the present devices that have been adapted for computing with force feedback studies are old robots or teleoperation manipulators. Because they have not been specifically designed for computer input/output application, problems associated with high mass, low frequency response, excessive friction, cumbersome size, high cost, and even safety concerns reduce their effectiveness. The electromechanical design of MagicMouse addresses these issues by featuring collocated sensing and actuation and is described hereafter.

Positioning Mechanism Structure

As can be seen in Figure 1, the main structure of MagicMouse is a stationary unit that incorporates a light weight 2-DOF moving plate used to direct the motion of a pointer on the screen. Two perpendicular linear ball-slides and a single low-friction, teflon leg make up the kinematic suspension system that constrains the plate to translational planar motion.[1] Alternative schemes, including a SCARA robot style configuration or a gantry system of teflon guides sliding on polished shafts, also have been experimented with. Since payloads are small, results have been positive.

Positional tracking of the plate is achieved by directing the light of an infrared light-emitting diode (LED), hidden inside the handle, onto a two-dimensional position sensitive detector (PSD) affixed to the base plate. The PSD output defines the handle location with respect to the center of the PSD. The physical motion range of this prototype is limited by the relatively small 1.7x1.7 cm^2 active area of the PSD. Other discrete or continuous position sensors could be employed, especially to increase motion range and reduce cost, as long as low-friction, end-point and collocated sensing is maintained.

Electromagnetic Actuator

Two electromagnetic flat coil actuators incorporated into the coil plate as seen in Figure 1 provide the tactile and kinesthetic feedback capabilities of the device. The L-shaped aluminum plate, attached to the base with three tubular supports, anchors two permanent magnet stators above each coil. In the same way, matching permanent magnet stators are anchored to the base below the coils.

Figure 2 is an exploded view of a single motor coil and its permanent magnet stator assemblies. When the prototype is assembled, the separation between the upper and lower magnets is set to about 5 mm with the 2.5 mm thick coil plate situated in the middle of the gap.[2] In the prototype, the coil plate consists of a hand-wound voice-coil that is sandwiched between two sheets of aluminum.

When a coil current I flows through the voice coil, it interacts with the magnetic field **B** and produces a force described by

$$\mathbf{F} = -I\int_C \mathbf{B} \times d\,\mathbf{l}$$

where $d\mathbf{l}$ is a differential element of wire in the coil, and the integration is performed over the entire wire. Considering that the field is roughly constant in the magnet gap (0.6 T) and negligible elsewhere, the net force on the coil acts linearly along one axis with its orientation dictated by the direction of the current. It follows that the x- and y-direction actuation forces on the coil plates can be controlled by two independent, bi-directional currents. Each coil has a measured resistance of 2.4 Ω and an inductance of 0.4 mH, giving coil time constants that, without current feedback, limit the device's force bandwidth to roughly 1 kHz.

Power MOSFETs, driven by pulse-width modulated signals (256 discrete-step, variable-duty pulses at 31.25 kHz), supply the excitation current to the coils. A force resolution of approximately 25 mN is achieved. A maximum current of 3 A yields a 6.3 N force.

The flat coil actuators were designed for uniform actuation over the full motion range. As can be seen in Figure 2 (coil is shown at its mid-motion range position in both DOF) a constant force output is obtained given a constant coil current, independent of the coil plate position within its specified 17x17 mm^2 motion range.

For future development, it is appropriate to consider how the dimen-

[1]Modified from a previously reported four-slide gantry.

[2]It is recognized that this leaves an unused air gap of 2.5 mm, which has an adverse effect on actuator performance, but allowed for experimentation with plates and coils of different thicknesses.

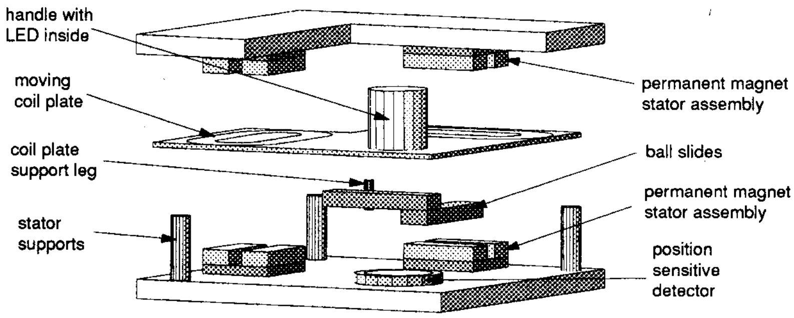

FIGURE 1: MAGICMOUSE PROTOTYPE ASSEMBLY – EXPLODED SCHEMATIC

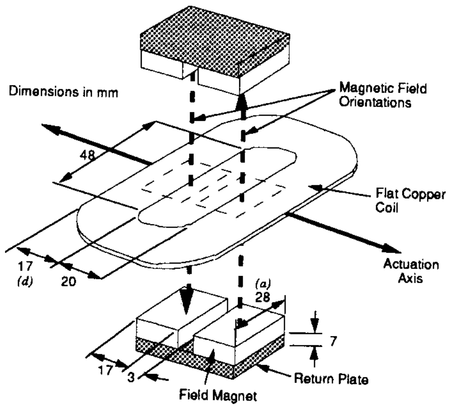

FIGURE 2: EXPLODED VIEW OF FLAT COIL ACTUATOR CONFIGURATION

sions of the entire device and its components would be scaled in an attempt to increase motion range. Experience with the prototype suggests that a motion range of approximately 40x40 mm^2 would be sufficient to allow control by one's fingers and wrist, without arm motion. No difficulties have been reported with the *Felix* input device (Bobker, 1988), which has a motion range of only 25x25 mm^2.

The coil size can be reduced by eliminating the non-conducting central "hole". The effect of this change is that the magnets now have to be separated so that they straddle the outside edges of the coil rather than the inside edges. This does not adversely affect performance, the "constant force" characteristic being preserved. With magnets of length a (a = 28 mm in Figure 2), and desired motion range of d in x and y, the total device area with such a configuration is approximately $A = (a + 5.2d)^2$. If the coils are moved closer together until the adjacent curved areas overlap, the total device area shrinks to approximately $A = (a + 4.5d)^2$. A 16 cm^2 motion range device of this configuration could be built to have a desktop footprint of approximately 21x21 cm^2.

A final scheme that has been considered for shrinking the device footprint while maintaining both linearity and maximum actuation force is to separate the coils into two plates connected by parallel links, one above the other. The permanent magnet stators would also be stacked on top of each other, yet both remain stationary. Although this configuration means that the suspension mechanism would have to be modified and the height of the device would increase slightly, the total device area becomes roughly $A = (a + 3d)^2$. A 16 cm^2 motion range device can be designed to have a footprint of 15x15 cm^2. For comparison, typical mouse pads have areas ranging from 17x20 cm^2 to 19x23 cm^2.

MAGICMOUSE INTERFACE AND CONTROL

Our haptic extension to a graphical user interface was implemented using the XView OPEN LOOK Toolkit for X Windows running on a Sun Microsystems SPARC station platform. XView was chosen because it can be customized and allows extensions to common user-interface objects.

Haptic senses require substantially higher "refresh" rates than vision. Indeed, it is now well accepted that for a "transparency" in telemanipulation, forces at the slave must be played back at the master to at least 500 Hz (Fischer et al, 1990). This implies that the haptic interface must be controlled at a sampling rate well exceeding 1 kHz.[3]

The workstation cannot perform the real-time control of the haptic interface for several reasons: First, the 1200 baud connection to the SPARC station mouse port would cause unreasonable latency times; second, code operating under X Windows (and Unix) cannot run in real-time; and third, software routines that control the user-interface have to coexist with other host applications. Because the user-interface is in a sense only window-dressing for other tasks, its implementation cannot and should not dominate the CPU time for the sake of force feedback. In comparison to the stringent timing that is associated with force feedback, the user-interface graphics do not have to operate as quickly, as video refresh rates as low as 15-30 Hz produce quality graphical illusions.

Therefore, a dedicated microcontroller CPU is employed to carry the computational load, in addition to performing position sensing and force actuation tasks. MagicMouse uses a 16 MHz Intel 80C196KC embedded microcontroller with on-chip timers, serial ports, 10-bit A/D converter, and pulse-width modulation (PWM) outputs. The microcontroller is also interfaced with a PC monitor that was used as a 'C' language development environment.

The microcontroller has two lines of 1200 baud, RS-232 serial communication with the SPARC station: a connection to the mouse port for supplying the workstation with mouse position data and a connection from the serial port for receiving supervisory information. The complete hardware configuration of the prototype system is shown in Figure 3.

[3]Note that no quantitative information on the subject has been reported in the literature, and this number is very much a consensus from qualitative reports. Higher force bandwidth requirements are often reported.

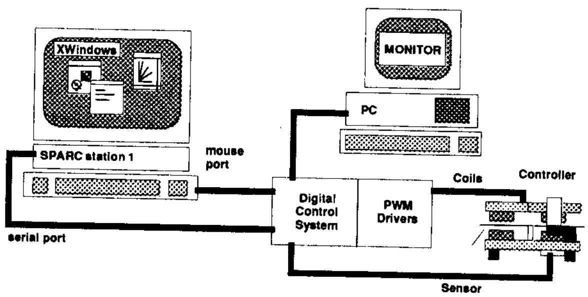

FIGURE 3: SYSTEM CONFIGURATION

MagicMouse is an absolute positioning device which means there is a one-to-one correspondence between the position of the controller and the position of the screen pointer. A rectangular subregion of the total PSD active area is quantized through software into 1152 columns and 900 rows and mapped to the 1152x900 pixels on the SPARC station display. The bordering secondary area of the PSD is used for sensing motion outside the display-mapped area and is necessary for the implementation of the force feedback scheme (discussed later) which lets the user 'feel' the edges of the display. Figure 4 shows the mapping between the sensor and display.

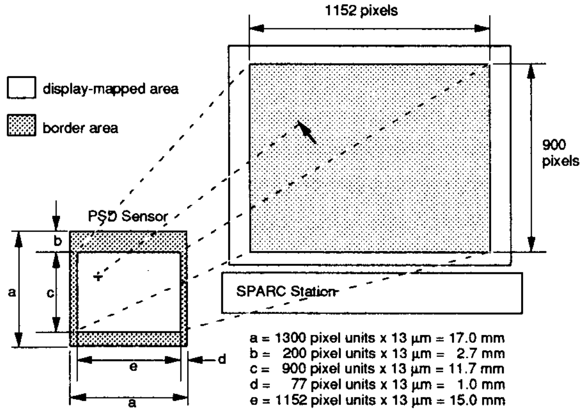

FIGURE 4: MAPPING BETWEEN PSD SENSOR AND DISPLAY

One important observation derived from these experiments and the use of PSD sensing is that for this type of haptic interface device, a high-resolution, low-accuracy sensor is just as good as a high-resolution, high-accuracy sensor, which is more difficult and expensive to build. The tetralateral type PSD we used suffers from a "barrel" distortion that causes skewed position readings, especially near the edges of the active area. In the case of MagicMouse, the display-mapped portion of the sensor would not be truly rectangular, but instead, the sides curve slightly towards the center. Our experiments have shown that this lack of accuracy is not perceptible.

Software/Hardware Integration

In the division of computational load, the host performs a supervisory role over the microcontroller across the 1200 baud communication channel. The microcontroller performs a digital sense-control-actuate loop and transmits mouse status data to the host's mouse port.

Central to the microcontroller software are two sets of position variables: (x_m, y_m), the position of the mouse when last sampled, and (x_p, y_p), the position of the pointer on the display. The microcontroller uses a polling-loop routine that initiates transmission of mouse position data, $(\Delta x_p, \Delta y_p) = (x_m, y_m) - (x_p, y_p)$, to the host.

The polling-loop can be interrupted by two main sources: supervisory communication from the host, and a software timer which launches sense-control-actuation routines. The PSD voltages are sampled and the mouse position (x_m, y_m) is calculated using linear interpolation.

Four generalized force feedback functions are employed for the haptic simulation of mechanical elements as will be discussed in the next section. When these routines are executed in sequence, they consider the current state of the display and modalities of the mouse, and calculate the forces by comparing that data to the present location of the mouse. A block diagram of the physical system including the user is shown in Figure 5.

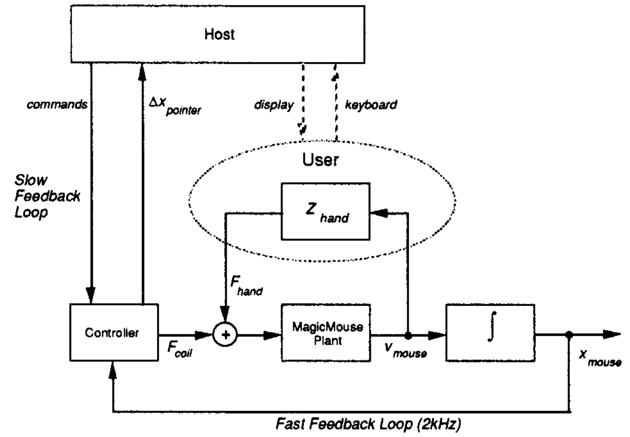

FIGURE 5: BLOCK DIAGRAM OF PHYSICAL SYSTEM

For our mechanical user-interface simulation experiments, the four A/D conversions of the sensor voltages take a total of 100 µs and control calculations take approximately 300 µs. The frequency of the control loop is set to 2 kHz rate, providing a 500 µs loop period sufficiently long for the 400 µs sense-control-actuate time.

Through the experiments discussed in the next section, it was found that the sampling rate could be lowered to a minimum of about 200 Hz before haptic simulations were seriously degraded. In this case, the device would respond to fast hand movement with rapid "punch-like" actuation that has a destabilizing effect. Results with prototype suggest that a preferred sampling rate would be 500 Hz or higher.

Table 1 summarizes the important characteristics of the MagicMouse prototype. A closed-loop position frequency response exceeding 30 Hz was measured using a dynamic signal analyzer. For this measurement, simple position control was employed at a sampling rate of 2 kHz and positional gain set to the same as that used for the mechanical slider simulation discussed in the next section. The peak amplitude of the noise stimulus position signal was about ±200 µm.

TABLE1. SUMMARY OF MAGICMOUSE PROTOTYPE CHARACTERISTICS

Desktop Footprint	14x14 cm^2
Mass (Total)	1.8 kg
Mass (Coil Plate/Gantry)	180g
Closed-loop Position Frequency Response	30 Hz
Coil Resistance	2.4 Ω
Coil Inductance	0.4 mH
Force Frequency Response	1 kHz
Magnetic Field in Gap	0.6 T
Maximum Coil Current	±3 A
Maximum Force	±6.3 N
Force Resolution	25 mN
Motion Range	1.7x1.7 cm^2
Display : Sensor Mapping	1 pixel : 13 μm
Control Rate	2 kHz
Sense-to-Actuation Latency	300-400 μs

MECHANICAL SIMULATION EXPERIMENTS

The goal of this research was to develop a high performance force feedback prototype that could be used to obtain a better understanding of the utility of a haptic enhancement to a graphical user-interface.

Mechanical simulation experiments using MagicMouse were motivated from a user's perspective. Primitives, considered to be important building blocks in a haptic /graphical user-interface hybrid such as that proposed in final section, were implemented as follows:

Mechanical Limit Stop Simulation

A spring force model was used to create a 'feel' similar to that of a mechanical stop or the striking of a hard surface. It is demonstrated by constraining the mouse movement to the areas defined by the display dimensions or the edges of an XView window.

If, for example, the mouse is pushed to the right when the pointer on the display is constrained by the right side of a window, the force actuated in the x-direction coil is depicted in the force profile of Figure 6, where the distance is the difference in pixel units between the pointer's location at the edge of the window and the actual location of the mouse. This distance can alternatively be viewed in terms of the controller's actual displacement in μm (one pixel : 13 μm).

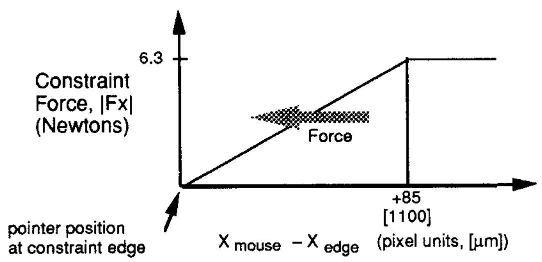

FIGURE 6: LIMIT STOP SIMULATION FORCE PROFILE

The force profile shown can also be regarded as a simulation of a physical spring with a very high spring constant, which gives the illusion of the mechanical stop. Discussions in Minsky et al and Ouh-young et al (1989) address this way of interpreting and modelling the simulations of hard surfaces, including contact instabilities which can occur. Some experimentation has shown that at a stiffness of approximately 11.5 N/mm our haptic simulation begins to feel "grainy", probably caused by signal quantization and noise.

Generally, because stiffness values below the instability threshold were found to be quite adequate in simulations of mechanical objects, including rigid mechanical stops or hard surfaces, little experimentation with stabilizing schemes (such as damping via negative velocity feedback) was attempted. The drawback of these techniques is that they can manifest themselves in a sluggish or spongy feel. Moreover, our experiments showed that the noise amplification resulting from numerical differentiation of the position signal detracts substantially from benefits that the addition of damping may have.

Mechanical Slider Simulation

The host was setup so that when an appropriate function key is pressed, simulation of a mechanical slider would be initiated. When the user tries to push the mouse in a direction perpendicular to the axis of the slider, the mouse exerts a force in the opposite direction creating the 'feel' of mechanical restriction. The force profile depicted in Figure 7 was very effective at stiffness values of 5.7 N/mm.

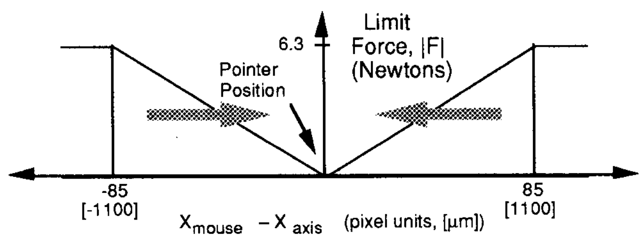

FIGURE 7: SLIDER SIMULATION FORCE PROFILE

Potential Field Simulation

Gravitational-type simulations were implemented with XView icons through the use a roughly inverse-distance model which offers increasing attraction as the pointer moves toward the icon.

The force profile shown in Figure 8 describes the attractive force on the pointer in the direction of the icon gravity sink, where the distance is the pixel unit or μm displacement between the pointer and the center of the icon. It reveals that as the pointer moves closer to the sink from afar, the attractive force increases linearly. As it moves closer still, it reaches a plateau range where the attractive force remains constant. Then, once the pointer gets very close to the center of the icon, the force is ramped back down to zero to create a kinesthetically stable location around the sink.

If the pointer is in the icon's vicinity where the attractive forces are greater than the frictional forces in the kinematic design, MagicMouse will move itself until the pointer and icon coincide.

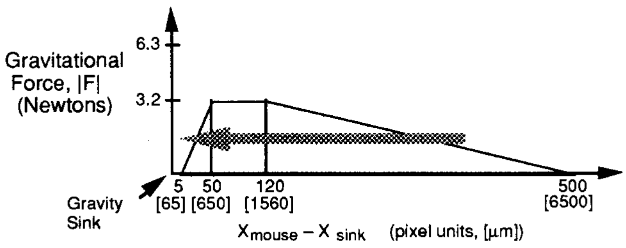

FIGURE 8: ICON GRAVITY SINK FORCE PROFILE

Mechanical Push-button Simulation

The final feature is probably the most intriguing since it better sparks the imagination of how more sophisticated force feedback interface objects might be created. This last feature is a push-button object that was created as a new XView widget package. The visual image of this widget is the graphic profile of a button that appears to move as though "pressed" when the pointer is pushed against it. In conjunction, MagicMouse simulates the force of an actual spring-loaded mechanical button.

The independent forces acting on the mouse in the x-direction for this push-button are defined by the force profile in Figure 9. The x-direction profile shows that as the button is pushed through its motion range, the resistive force that pushes outward is linear with distance. The discontinuity at the end of this range produces a tactile clicking sensation. When the button is fully pressed, the final force ramp indicates a mechanical stop similar to that of Figure 6. Initially, it was found that the pointer "slipped off" the surface of the button in the y-direction too easily as the user pushed the push-button. To correct this, the y-axis is effectively turned off upon touching the push-button with a force characteristic the same as in Figure 7.

Although the approximately 1.25 cm stroke of the graphical button presentation on the screen is vastly greater than the 430 µm stroke of the force profile actuated by the controller, experiments have shown that this scale difference between the visual and kinesthetic presentations does not cause a cognitive conflict for the user nor reduce his "belief" in the simulation.

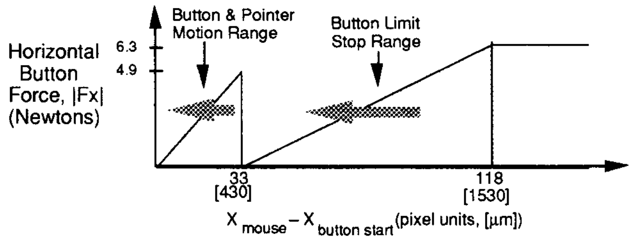

FIGURE 9: PUSHBUTTON USER-INTERFACE OBJECT FORCE PROFILE

USER-INTERFACE APPLICATIONS

Through experimentation with MagicMouse and the demonstrational force feedback system described previously, ideas for a haptic/graphical user-interface hybrid were developed and are proposed in this section. However, future studies may either prove or disprove their real utility.

Figure 10 shows an example of a graphical user-interface that could be augmented with a tactile and kinesthetic interface. The short black arrowheads indicate the directions in which force feedback could be used to enhance the user's ability to function. The following is a description of the interface elements referenced by letter in Figure 10:

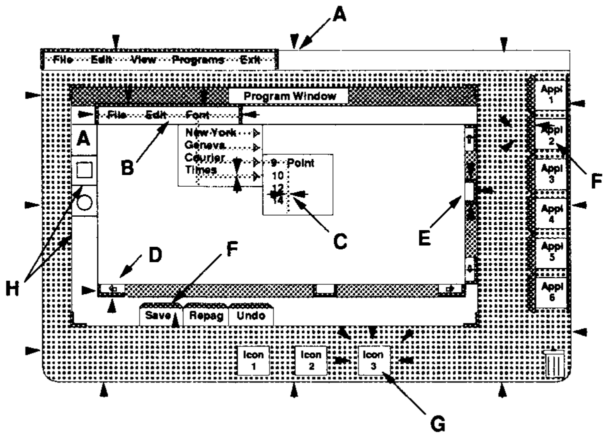

FIGURE 10: IMAGE OF GRAPHICAL USER-INTERFACE INCORPORATING HAPTIC FEEDBACK

A Rectangular Constraints

As demonstrated previously, force borders can be created at all four edges of the display's workspace. When the pointer is moved to a corner, the user can feel the corresponding kinesthetically stable corner in the controller's motion range, and is constantly given a sense of the workspace bounds. If relevant, the user could be also confined to a smaller rectangular area such as a window.

B Menu Bars

Users want to quickly locate desired menus with as little cognitive effort as possible. This task can be made easier by making the menu bar a kinesthetically stable place when it is approached from below. In this way, when the pointer is moved rapidly in the direction of the menu bar from below, force feedback is used to impede mouse movement and the pointer's progress past the menu bar.

Once at the menu bar, force feedback continues to aid the user in arriving at the desired title in that the pointer is allowed to move only in the left, right, and down directions when on a menu bar. Thus, movement of the pointer to the left or right along the menu bar is less demanding on the user's motor skills than if a similar operation is done with a normal mouse because MagicMouse prevents accidental slippage off the menu bar as the user affects lateral movement. This is a significant improvement over normal mice in which a slight rotation of the mouse in the user's hand creates coupling between x and y axes, violating the user's sense of direction (Williams, 1989).

Force feedback could also be used to select a menu title as an alternative to pressing the mouse button, by employing a force threshold in the upward direction that would indicate selection.

C Menus

After a menu title has been selected from a menu bar, a menu would drop down. Since picking an item from a menu requires movement of the pointer in a vertical direction, the mouse could be used to "turn off" the horizontal direction of motion as demonstrated previously.

Figure 10 shows how force feedback could be used to enhance the hierarchical menus that are common today. The lightly shaded lines depict the paths to which the controller's motion would be restricted as the user negotiates the menu. These axes act like mechanical tracks that can help the user traverse the menu tree and arrive at the final desired selection by eliminating the troublesome x and y motion coupling of regular mice. Force feedback could also be used to let the user feel the "jump" from one menu item to another.

D Scroll Bar Arrows

Force feedback can be used in conjunction with the arrows at the ends of scroll bars to accelerate the scrolling process by allowing the user to move faster and be more carefree when trying to position the pointer over the arrow object. This can be done through the creation of kinesthetically stable sides on the arrow object that prevent the pointer from over-shooting. These forces would actually impede the motion of the mouse as it is moves across the target.

Force feedback can be used as an alternative to pressing the mouse button after the pointer is positioned over the arrow object. Scrolling would be facilitated whenever the exertion force exceeds a threshold. It would also be possible to link scroll speed to force levels.

E Scroll Bar Thumbs[4]

As with scroll bar arrows, force feedback can be used to create kinesthetically stable sides on the thumb object to prevent pointer overshoot. In the example referred to in Figure 10, the controller would constrain the pointer from going beyond the right side of the thumb after it has "entered" it from the "open" left-hand side.

MagicMouse could be programmed to allow the user to move the thumb without the need for pressing a button. To do this, if the pointer is pushed against the top or bottom of the thumb once it has been "entered", the thumb would follow the motion of the pointer. As the thumb is moved, a viscous drag sensation could be simulated in the motion direction When the thumb has reached the limit of its motion range, this too would be reflected via the use of forces.

F Push-buttons

In many applications, some frequently used commands are inconvenient to access from menus. An alternative access could be to employ user-programmable push-buttons, as shown in Figure 10, that are similar to those demonstrated. The push-buttons would eliminate the need for using the button on the mouse and still provide a tactile feedback similar to that in a mechanical button. Similar push-buttons could be arranged in a panel as a means to launch applications. A force gravitation scheme could also be used to aid the user in arriving at any of the buttons with the least possible effort.

G Icon Gravitation

The icon gravitation that was previously demonstrated gives another example of how force feedback can increase positioning speed. This could be of great use to the visually impaired or those lacking strong motor skills by enabling them to more easily move the pointer to unambiguous positions. The benefit of kinesthetically stable locations such as these have been demonstrated with *Felix* (Williams, 1989).

H Window and Region Boundary

As the user moves the pointer across a window or region boundary, the user could be given a tactile signal of that fact. A small instantaneous force, normal to the region edge, could be actuated to indicate that the window has been entered or exited.

A variety of other possibilities for the force feedback interface exist. Viscous damping can be used to give the user an intuitive feel of dragging a selection. Pointer gravitation could be useful when modal dialog boxes popup on the screen by forcing the user to attend to them.

---

[4]Scroll bar thumbs, sometimes called scroll boxes, facilitate scrolling when dragged with the pointer.

As seen before, one promising MagicMouse feature is the ability to arbitrarily turn off degrees-of-freedom, preventing cross-coupling between motions between axes. In a study of 3 DOF input devices for CAD applications by Robert Beaton and discussed in Williams (1989), it was shown that traditional plane-oriented (2 DOF) and free-space-oriented (3 DOF) devices are not as popular as vector-oriented devices such as thumbwheels because they are often not as fast or lack the same positioning accuracy. The reason for this is that while the former devices do not possess this advantageous axial independence characteristic, the latter devices do.

Given this, it becomes clear that one of the salient features of the MagicMouse is that it can be used for gross movements through two-dimensional space towards a destination in a much more intuitive way than would be possible with a vector input device. Then, once the pointer is close to the destination, one or more axes of the controller's motion can be effectively turned off to facilitate accurate positioning if necessary, a task that regular mice cannot do.

CONCLUSIONS / FUTURE WORK

A new generation of input /output devices that the exploit the human sense of touch are fundamental to the progressive expansion beyond purely graphical user-interfaces toward a more holistic visual-haptic hybrid. The prototype MagicMouse device discussed in this paper has demonstrated that the force feedback technology needed to bring the sense of touch to the human-computer interface is available and can be produced at relatively low cost.

A novel electromechanical design has been successfully integrated with a host workstation through hardware and software. Experimental results of this first reported and demonstrated mechanical force feedback enhancement of a graphical user-interface using MagicMouse have been discussed. Guidelines for future user-interface designs and integration have been given. It is also regarded that in future MagicMouse designs, issues relating to the improvement of sensor and kinematic designs, increasing of motion range, and the development of a better host to mouse hardware interface will be central to further development. Additionally, it is hoped that a third (full or half) translational degree-of-freedom may be added to the planar prototype device to increase functionality. Finally, a systematic psychophysical study may have to be completed to confirm that the addition of force feedback to the user-interface is indeed useful beyond novelty.

REFERENCES

Atkinson, W.D., Bond, K.E., Tribble, G.L., Wilson, K.R., 1977 "Computing with Feeling," *Computers and Graphics*, Vol. 2, pp. 97-103.

J.J. Batter, J.J., Brooks, Jr. F.P., 1972, "GROPE-1: A Computer Display to the Sense of Feel," *Proceedings of the International Federation of Information Processing (IFIP)*, pp. 759-763.

Bobker, S., 1988, "Felix," MacUser, September, pp. 79-80.

Davis, B., 1990, "Illusions," *Discover*, June, pp. 37-41.

Fischer, P., Daniel, R., Siva, K.V., 1990, "Specification and Design of Input Devices for Teleoperation," *Proceedings of the IEEE International Conference on Robotics and Automation 1990*, pp. 540-545.

Fisher, S., McGreevy, M., Humphries, J., Robinett, W., 1987, "Virtual Interface for Telepresence Applications," J.D. Berger (ed.), *Proceedings of ANS International Topical Meeting on Remote Systems and Robotics in Hostile Environments.*

Foley, J.D., 1989, "Interfaces for Advanced Computing," *Scientific American*, October, pp. 127-135.

Hollis, R.L., Salcudean, S.E., Allan, A.P., 1991, "A Six-Degree-of-Freedom Magnetically Levitated Variable Compliance Fine-Motion Wrist: Design, Modeling, and Control", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, June, pp. 320-332.

Hannaford, B., Szakaly, Z., 1989, "Force-Feedback Cursor Control," *NASA Tech Briefs*, Vol. 13, No. 11, Item #21, November.

Iwata, H., 1990, "Artificial Reality with Force-feedback: Development of Desktop Virtual Space with Compact Master Manipulator," *Proceedings of SIGGRAPH '90*, Dallas, August 6-10, pp. 165-170.

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

Noll, A.M., 1972, "Man-Machine Tactile Communication," *J. Soc. Inform. Dis.*, July-August.

Ouh-young, M., Minsky, M., Behensky, M., Brooks, Jr. F.P., 1989, "Creating an Illusion of Feel: Control Issues in Force Display," Computer Science Department, University of North Carolina at Chapel Hill, September 16.

Ouh-young, M., Pique, M., Hughes, J., Srinivasan, N., Brooks, Jr. F.P., 1988, "Using a Manipulator for Force Display in Molecular Docking," *IEEE International Conference on Robotics and Automation*, pp. 1824-1829.

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

Schwartz, R., 1988, "Alternative Mice Styles," MacUser, July, pp. 198-205.