Haptic Pen: A Tactile Feedback Stylus for Touch Displays

Johnny C. Lee[1,2], Paul H. Dietz[2], Darren Leigh[2], William S. Yerazunis[2], Scott E. Hudson[1]

[1]Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{ johnny, scott.hudson}@cs.cmu.edu

[2]Mistubishi Electric Research Laboratories
201 Broadway St
Cambridge, MA 02139, USA
{dietz, leigh, yerazunis}@merl.com

ABSTRACT

In this paper we present a system for providing tactile feedback for stylus-based touch-screen displays. The Haptic Pen is a simple low-cost device that provides individualized feedback for multiple simultaneous users and can operate on large input displays as well as ordinary surfaces. We combine a pressure-sensitive stylus with a small solenoid to generate a wide range of tactile sensations. The physical sensations generated by the Haptic pen can be used to enhance our existing interaction with graphical user interfaces as well as to help make modern computing systems more accessible to those with visual or motor impairments.

KEYWORDS: haptic feedback, stylus, touch screen, multiuser

INTRODUCTION

Touch-sensitive surfaces and stylus-based tablets have become a common interface technology for modern computing devices. These input technologies are often spatially coupled with a display to offer us a sense of direct manipulation with screen objects. We find these input displays used in hand-held devices, tablet PCs, and large collaborative work displays. By unifying the location of inputs and outputs, they reduce the disconnection between action and reaction found with other input devices such as mice. However, these interface renderings are still far from seeming realistic.

Graphical interfaces on an input display lack the physical response that our highly developed visual-motor systems have come to expect from actions such as pressing a button. This lack of feedback creates a discontinuity between expectation and experience causing our interaction with the interface to take a step back towards the abstract.

To improve this situation, we have developed a lost-cost method for providing an approximation of these physical sensations through tactile feedback for stylus-based input displays. Our design uses a pressure sensitive stylus in combination with a locally mounted physical actuator. By placing the actuator in the stylus, we are able to support multiple users simultaneously, provide uniform feedback quality regardless of screen size or geometry, and provide tactile feedback even when the user is not actively pressing on the display surface.

Submitted to UIST '04

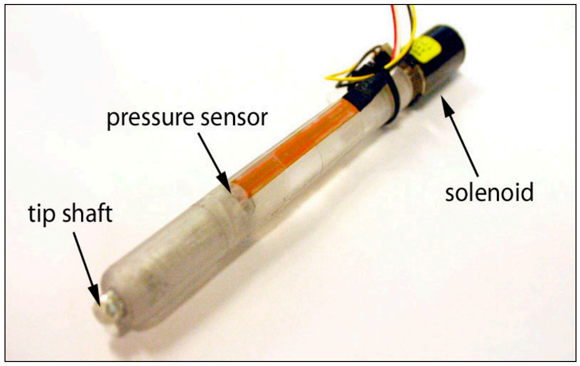

Figure 1. Tactile feedback stylus

RELATED WORK

Providing tactile or haptic feedback for graphical user interfaces has been explored for some time, particularly in the field of assistive technologies and rehabilitation engineering. This work has focused on making modern computing systems more accessible to those with motor or visual impairments [Way97][Yu00][Sjo99]. Other research has explored the benefits of tactile feedback in computer interfaces for all users [Bra01][Cha02]. Technologies used in this work have included SensAble Systems' Phantom Haptic Device [Oak00], vibration-capable mice [Gob95][Ter00], or fully tactile displays using large actuator arrays. However, most of these technologies are inappropriate for use with touch-sensitive or tablet-based displays.

Providing tactile feedback for touch screens [Pou03][Fuk01] has previously been achieved by placing a physical actuator directly behind the touch surface of the display device. This technique is effective for small devices such as PDAs or palm-top computers, but does not scale well to larger screen sizes. Additionally, this technique cannot provide individualized feedback for multiuser systems.

By placing the physical actuator in the stylus rather than the display, we can remove these limitations as well as gain several new benefits. We can detect tip pressure, utilize location data while “hovering”, and maintain a constant tactile communication channel to the user.

THE HAPTIC PEN

Our haptic pen design is a simple, low-cost method for providing tactile feedback that can enhance our interaction with graphical user interfaces. First, because the actuator in the stylus requires a power supply, we assume an active stylus system. Active stylus designs are fairly common for large projected displays as well as passive display technologies, such as pens with *Anoto Functionality* [Ano02], which uses special-purpose paper. Secondly, we do not attempt to generate reflective forces that resist stylus movement, (e.g., for stopping movement at simulated boundary edges). This approach requires an armature system, such as the Phantom, which increases both the cost and complexity of the feedback device tremendously. We feel the simplicity and affordability of our approach results in a design that is very practical.

To create a haptic pen, we need five components: a physical actuator, a pressure-sensitive tip, a location-discovery system, a communication link with a host PC, and a source of power. Most of these components are already available in existing active-stylus touch-screen technologies.

The choice of the physical actuator is critical to the effectiveness and expressiveness of the tactile feedback. To better understand what will make an effective actuator, we must first look at the forces involved in an interaction. When a user presses a button, the resulting force vector is primarily aligned with the longitudinal axis of the stylus. Therefore, generated reaction forces should also be directed along the longitudinal axis of the stylus to create a coherent tactile experience. An actuator that produces substantial lateral forces, such as an eccentric mass vibrator, will produce a less-convincing effect since the reaction will largely be perpendicular to the action. Additionally, the actuator must be capable of delivering high-energy impulses without oscillation to mimic the sudden forces of a button.

We explored several actuators that meet these requirements including linear actuators, piezo stack actuators, and solenoids. Though each technology has its own merits, we found that a solenoid provided the best overall solution in terms of cost, size, force, reaction speed, and expressive capabilities.

We use a small push-type solenoid mounted coaxially at the “eraser” end of the stylus. The shaft of the solenoid is rigidly attached to the stylus body and the coil housing acts as the actuated mass. This keeps the stylus design mechanically very simple, which increases physical robustness and eases manufacturing. Accelerating the mass away from the tip toward the rear of the stylus to a hard limit-stop generates the primary force. A secondary force comes from allowing gravity to pull it back down to its rest position. It is important to note that the primary force is

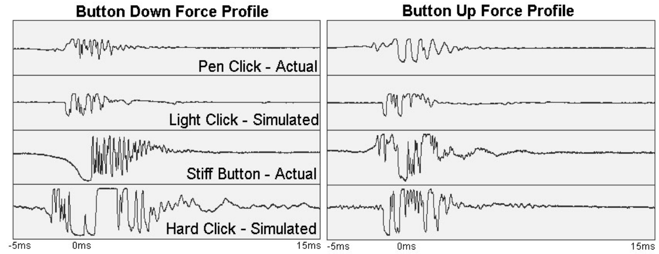

Figure 2. Accelerometer data comparing actual and simulated forces, left – button down, right – button up.

directed away from the direction of the display surface and the tip stays in constant contact held under pressure from the user. Otherwise we could cause damage by effectively hammering the tip into the screen. The tip remains stationary with respect to the stylus.

The solenoid (Guardian Electric model A420-067074-01) has a 16.1mm diameter, and provides an actuated mass of 26.7g (36g total mass). With a 20V power supply, it is possible to generate about 50mJ of impact energy within 5ms. The energy for this kick can be delivered by a 100µF capacitor if a high-current power supply is not available. Once the solenoid is in the lifted position, less than 1mA of sustained current is necessary to hold it up, sufficiently low for battery operation.

We implement a simple pressure-sensitive tip using a metal shaft insert, which transfers the tip pressure to a variable-resistance compression sensor (CUI model SF-5) placed inside the stylus. The metal tip provides a conductive channel through the stylus for capacitive sensing with a DiamondTouch table [Die01]. Though any touch technology can be used, the DiamondTouch table supports touches by multiple users simultaneously. One of the benefits of the haptic pen design is its ability to provide individualized feedback in a multi-user setting.

The control circuit uses a PIC16F876 microcontoller, which controls the solenoid, digitizes the pressure sensor, and communicates with the host PC with a very small number of additional components. The microcontroller has a built-in 10-bit A/D converter, pulse-width modulation hardware, and RS-232 capabilities. Low-level control routines are handled by the micro-controller, while the PC control software selects which overall haptic behavior is desired.

We used an Analog Devices ADXL202 accelerometer to examine the force profiles generated by the pen when executing different haptic behaviors. This data, shown in Figure 2, clearly show similarity both in terms of duration and overall profile shape when compared to an actual retractable pen and a stiff mechanical button. Many of the residual differences are in the 1kHz range and approach the limits of human perception [Cho95]. This shows that our simple solenoid-based design is capable of producing sensations similar to familiar mechanical switches. The total cost of the components in our prototype pen was less than $10.

HAPTIC BEHAVIORS

We treat a haptic behavior as a set of physical actions that have been mapped to states and transitions. Transitions between states are conditional upon input from the user. An appropriate selection of actions will define a coherent tactile rendering of a physical control such as a button. Figure 3 illustrates the action diagram for a "*Basic Click*" behavior. Each state and each transition has an associated solenoid control action (e.g., *off*, *lift*, *hold*, and *drop*). State transitions are taken when the conditions from a given state are satisfied (e.g., *tip pressure* > *down threshold* | state=*button up* → state=*button down*).

We currently have 5 basic solenoid actions: *Off*, *Hold*, *Lift*(*strength*), *Hop*(*strength*), and *Buzz*(*strength*). *Hold* is a low-power drive signal that keeps the solenoid in the lifted position. *Lift* generates a PWM signal that accelerates the mass upwards at a specified strength. *Hop* injects a single pulse that momentarily lifts the solenoid a specified amount before letting it drop back down to rest position. This can produce a sensation ranging between subtle clicks to heavy thumps. *Buzz* oscillates the solenoid drive signal to vibrate the mass at a specified strength. Creatively combining actions, selecting transition thresholds, and choosing strength parameters can yield a variety of distinct haptic behaviors.

BEHAVIORS AND GUI APPLICATIONS

To demonstrate the versatility of our haptic pen, we designed eight distinct behaviors within the space of haptic buttons. They are: *No Click*, *Light Click, Basic Click, Hard Click, Buzz, Force Buzz, Two-Click,* and *Buzz-Click*. We will discuss feedback behaviors for other GUI elements in a later section.

No Click provides no haptic feedback but generates the mouse down and up events so the visual feedback of the GUI is still rendered.

Light Click, *Basic Click*, and *Hard Click* simulate buttons of various stiffness using variations of the action diagram shown in the top of Figure 3. The differences are in the transition thresholds and actions. *Light Click* uses very low thresholds and *Hop(Light)* actions on both transitions creating the illusion of an easy to press button with light feedback when pressed and released. This effect is subtler than *Basic Click*, which uses higher thresholds to perform a medium-strength solenoid *Lift(Medium)* followed by a *Drop*. *Hard Click* creates the illusion of an extremely stiff button by using very high thresholds and a *Lift(Max)* action. This requires the pen to be pressed down very heavily before responding with a strong kick. The sensation produced by this behavior has been likened to using a powered punch tool or a small staple gun. The force profiles for *Light Click* and *Hard Click* are shown in Figure 2. The difference between these click behaviors is dramatic and showcases the wide range of physical expressions the pen can achieve. Clever assignments of these behaviors to GUI elements can add an affective component to our

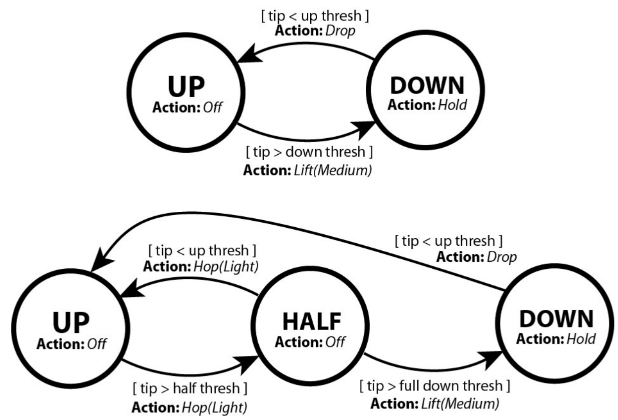

Figure 3. Action diagrams for *Basic Click* (top) and *Two-Click* (bottom)

interaction. Our bodies respond in a visceral manner to the tactile properties of objects. For example, a settings dialog may apply light feedback for each individual option but the confirmation button may be very stiff, requiring confidence in action from the user and possibly providing a sense of closure and completeness.

Buzz is a simple behavior that produces a mild buzzing sensation when the haptic button is depressed. Buzzing can indicate that an error has occurred, such as missing a specified target or attempting invalid input. *Force Buzz* changes the strength according to tip pressure. Pressing harder increases the buzzing strength.

Two-Click provides a two-level button similar to the shutter button on a still camera or [Zel01]. This is accomplished with the action diagram shown in bottom of Figure 3. When pressed halfway, the user receives a light-click sensation followed by a stronger full click if pressed harder. *Buzz-Click* is similar, but provides a buzz when pressed halfway. These multifunction buttons can combine related operations into a single graphical control. Two-level taps are also an elegant method of providing single-click and double-click operations in a single pen-down action.

Since the behaviors are controlled by software, haptic buttons can dynamically change their behavior to communicate information to the user. Toggle switches are examples of mechanical controls that change their physical qualities depending on the state of the application. We can simulate this behavior be alternating between *Light Click* and *Hard Click* feedback behaviors. Another example might be a "Check Email" button that becomes stiffer depending on the quantity of new mail. By feeling for stiffness, the user is able to "peek" at the data behind a button without having to commit to its execution.

BEYOND THE BUTTON

Thus far, our haptic behavior exploration has primarily focused on button simulation because the components of button interaction encompass most of our interaction with GUIs. Some behaviors such as *Light Click*¸ *Basic Click*, *Hard Click* and *Two-Click* can be generally applied to most graphical interface operations. However, we found that the needs of dragging interactions are more varied and task dependent than simple buttons. It is difficult to select a haptic behavior that is uniformly appropriate. However, the Haptic Pen provides an expressive vocabulary that the designer can tailor to their needs.

An active pen design also allows us to obtain location data even when the pen is not depressed on the screen. This data can be used to drive haptic behaviors to aid GUI navigation. For example, Buzzing strength can be driven by proximity, region, or direction to guide users toward a target area. We can also signal the crossing of important edges with a variety of different thumps.

Since the stylus is held in the user's hand throughout the interaction regardless of contact with the display surface, we have an additional persistent channel of communication with the user. Though simple, this haptic display may be valuable when effective visual or audio feedback is impossible. In certain applications, tactile feedback has been shown to be five times faster than visual feedback [Gel60]. By varying click count, click strength, buzz strength, and duration, it may be possible to transmit a substantial amount of information to the user.

The Haptic Pen is also compatible with nearly any location-discovery technology and does not necessarily need to be used on an active display surface. For example, a six degree-of-freedom motion tracker allows any object with known geometry to be transformed into an input surface with haptic feedback. The pen could be used with a paper print out of a numeric keypad taped to a desk. Or more imaginatively, each black pentagon on the surface of a soccer ball could be defined as a different haptic button. An implementation using the Anoto pen location technology [Ano02] would allow you to draw a haptic button on paper with the pen itself and then press it as if it physically existed.

DISCUSSION AND FUTURE WORK

We informally invited 10 colleagues unfamiliar with the project to evaluate the strength of the illusion created by the haptic pen using a GUI containing the eight different haptic buttons, which included the *No Click* behavior for comparison. Eight of the individuals reported a strong belief that the pen was actually moving down with each button press. The remaining two individuals closely observed the solenoid movement, which made them unsure about their interpretation of the feedback. Enclosing the solenoid inside the stylus may reduce this confusion, strengthening the illusion. The qualitative feedback from these sessions was overall very positive indicating the illusion was convincing and interacting with this simple interface of 8 buttons was enjoyable. However, a more formal user study would be needed to determine if there are performance improvements or increased levels of satisfaction with the interaction.

Our next goal is to construct more mature prototypes that conceal the solenoid, package the control electronics to fit inside the stylus, and eventually work toward creating a wireless version with a local battery. Since we developed the pen to support multiple simultaneous users working on large input displays, our future development paths will focus on creating software and hardware technology for multi-user haptic applications.

REFERENCES

[Ano02] Anoto, "Development Guide for Service Enabled by Anoto Functionality", 2002: Anoto.

[Bra01] Brave. S., Nass, C., Sirinian, E., Force Feedback in Computer Mediated Communication. *Proceedings at HCI International 2001 Conference* (New Orleans, LA).

[Cha02] Chang, A., O'Modhrain, S., Jacob, R., Gunther, E., Ishii, H., "ComTouch: design of a vibrotactile communication device." In the Proceeding of the conference on Designing Interactive System, 2002.

[Cho95] Cholewiak, R.W. & Collins, A. A. "Vibrotactile pattern discrimination and communality at several body sites." Perception & Psychophysics, 57, pp. 724-737.

[Die01] Dietz, P.H.; Leigh, D.L., "DiamondTouch: A Multi-User Touch Technology", ACM Symposium on User Interface Software and Technology, November 2001

[Fuk01] Fukumoto, M., Toshaki, S., "ActiveClick: Tacile Feedback for Touch Panels." in CHI 2001, Extended Abstracts, ACM Press. p. 121-122.

[Gel60] Geldard, F. "Some neglected possibilities for communication." Science, 1960. 131 (3413): p.1583-1588

[Gob95] Göbel, M., Luczak, H., Springer, J., Hedicke, V., Rötting M. (1995) "Tactile feedback applied to computer mice". International Journal of Human-Computer Interaction, 7 (1), pp. 1 - 24

[Pou03] Poupyrev, I. and Maruyama, S.(2003). "Tactile Interface for Small Touch Screens". In proceedings of ACM UIST 2003, ACM Press.

[Oak00] Oakley, I., McGee, M.R., Brewster, S. & Gray, P. (2000) Putting the Feel in 'Look and Feel', In Proceedings of ACM CHI 2000, The Hague, ACM Press.

[Sjo99] Sjöström, C., Rassmus-Gröhn, K. The sense of touch provides new interaction techniques for disabled people, *Technology & Disability* Volume 10, Number 1, IOS Press, 1999.

[Ter00] Terry et al., "Tactile Feedback In A Computer Mouse," Proceedings of Fourteenth Annual Northeast Bioengineering Conference, University of New Hampshire, Mar. 10-11, 1988.

[Way97] Way, T., Barner, K. "Automatic Visual To Tactile Translation, Part I: Human Factors, Access Methods and Image Manipulation." IEEE Transactions on Rehabilitation Engineering, vol. 5, pp. 81-94, Mar. 1997.

[Yu00] Yu, W., R. Ramloll, and S. Brewster. Haptic Graphs for Blind Computer Users. in Haptic Human-Computer Interaction Workshop. 2000. Glasgow University.

[Zel01] Zeleznik, R., Miller, T., Forsberg, A. "Pop through mouse button interactions". In Proceedings of ACM symposium on user interface software and technology, 2001. ACM Press.