Soap: a Pointing Device that Works in Mid-Air

Patrick Baudisch, Mike Sinclair, and Andrew Wilson
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
{baudisch,sinclair,awilson@microsoft.com}

ABSTRACT
Soap is a pointing device based on hardware found in a mouse, yet works in mid-air. Soap consists of an optical sensor device moving freely inside a hull made of fabric. As the user applies pressure from the outside, the optical sensor moves independent from the hull. The optical sensor perceives this relative motion and reports it as position input. Soap offers many of the benefits of optical mice, such as high-accuracy sensing. We describe the design of a soap prototype and report our experiences with four application scenarios, including a wall display, Windows Media Center, slide presentation, and interactive video games.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. Input devices and strategies; B 4.2 Input Output devices

Keywords: Soap, input, pointing device, mouse, hardware, optical sensor, mid-air input, 10 foot user interfaces.

INTRODUCTION
A wide range of application scenarios require users to control an appliance while standing or walking, such as 10-foot user interfaces, wall displays and projected displays, and augmented reality applications. To handle these application scenarios, several mid-air pointing devices have been proposed. Examples are joystick-based presenter tools, game controllers, gyro mice, accelerometer-based devices, and magnetic trackers.

The mouse, unfortunately, *is not* a mid-air pointing device. Since it requires a surface to operate, it relegates users to a table and excludes them from the scenarios listed above.

A closer look reveals why the mouse does not work in mid air. All input devices, including the mouse, consist of two parts, i.e., the part that users move or apply force to and some sort of reference element. In the case of the mouse, that reference element is the surface the mouse is operated on, typically a table or a mouse pad. Moving the mouse above the surface causes it to lose that reference system. Without it, the mouse is only half an input device and not functional.

UIST'06, October 15–18, 2006, Montreux, Switzerland.

Figure 1: The core of soap is an optical sensor. The core can rotate freely inside soap's elastic hull. While pointing, users cause soap's core to rotate by applying off-axis pressure. The relative motion between core and hull is picked up by the optical sensor inside the core.

In this paper, we want to reclaim the mid-air space for the mouse, or at least for a variation thereof.

SOAP
Soap is an input device that is based on an optical sensor as used in optical mice. Yet, soap takes its "mouse pad" with it and can therefore be operated in mid-air.

Figure 1 shows a soap prototype and illustrates the design. Soap consists of two main elements. The *core* is a roughly lentil-shaped wireless device that contains an optical sensor facing outwards. The *hull*, which consists of elastic fabric, encloses the core. Any relative motion between core and hull is picked up by the optical sensor in the core and reported wirelessly to the appliance soap is connected to, such as a PC.

Since the core is completely surrounded by the hull, users cannot touch the core or move it directly. Instead, users operate soap by applying pressure from the outside as shown in Figure 2. Given its particular shape, the core evades pressure, which causes it to rotate independently of the hull. The resulting relative motion between core and hull is reported to the appliance.

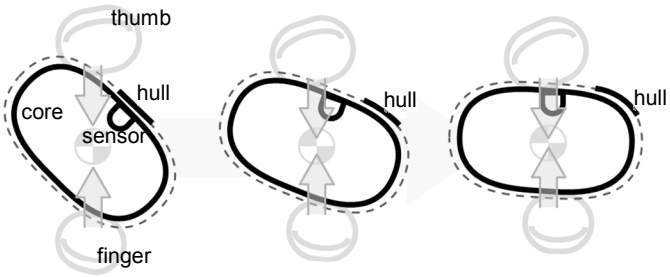

Figure 2: The mechanics of soap: As the user applies pressure using thumb and finger, the core rotates to evade the pressure. At the same time, friction between finger and hull holds the hull in place, resulting in relative motion between hull and core.

Figure 3 shows the core of one of our prototypes. It is based on a circuit board from a wireless optical mouse that we rewired to fit a 2x2x1" form factor (5x5x2.5cm). We use a plastic casing to give the device the required shape. The use of a clear casing allows the optical sensor to see the hull.

The shown model allows users to perform a click operation by squeezing the device, which is implemented by placing a micro switch appropriately inside the slightly flexible casing. Other prototypes we made feature additional buttons on the top and on the side of the device. Soap's buttons require pressure comparable to the buttons on a regular mouse. This pressure threshold is high enough to prevent accidental clicks during pointing interactions.

Since circuitry and wireless connector stem from a regular wireless USB mouse, soap is recognized by PCs as a plug& play mouse device. However, the specific design of soap causes its resulting mouse movements to be mirrored. A simple "MouseMirror" program we wrote rectifies this by intercepting and inverting mouse move events using a Windows "mouse hook".

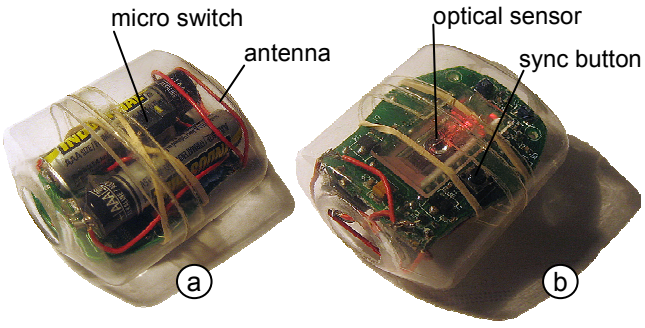

Figure 3: The soap core (a) top & (b) bottom view. A micro switch in a slightly elastic casing allows users to generate a click by squeezing the device.

The hull plays a crucial role in the design of soap. It guarantees reliable tracking and prevents the sensor from picking up the unintentional or erratic motion that occurs when moving a finger directly over an optical sensor. We use a hull that is elastic enough to fit tightly around the core. To insure an effortless interaction, we separate the hull from the core by an additional two layers that hold a lubricant (oil, soap water, or glycerin).

The hull also enables soap's three interaction styles.

Three types of motion

Soap supports three types of motion. Since the device is roughly as long as it is wide, all three interaction styles can be performed in x and in y, allowing users to point freely in 2-space.

Users perform a **joystick interaction** by dragging the fabric on top of the sensor using their thumbs (Figure 4a). The rest of the hand holds the device firmly, so that the hull stretches when tracking with the thumb. When the user releases the fabric, the hull returns to its original position. This type of self-centering behavior is reminiscent of a joystick, hence its name. Joystick interaction is soap's fastest and most precise interaction style.

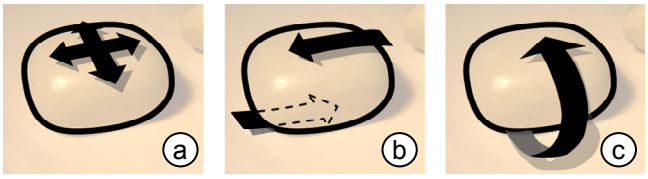

Figure 4: Soap supports three interaction styles: (a) joystick, (b) belt, and (c) soap.

Like the mouse, soap is a position input device [5], i.e., any motion over the sensor affects the *position* of the pointer, rather than pointer *speed* (as is the case for rate-controlled devices, such as joysticks). Like any position input device, soap needs to provide a mechanism for users to move across longer distances. Some devices, such as the mouse, offer a clutching motion; other devices, such as a dial, offer infinite motion. Soap offers both; they are called *belt* and *soap* interaction.

Users perform a **belt interaction** by dragging the fabric on top of the device in one direction and the fabric at the bottom of the device in the opposite direction (Figure 4b). Users prevent the device from flipping over by applying some vertical pressure. The belt interaction allows users to position a pointer without the self-centering behavior of the joystick interaction. Repeated belt interactions allow users to move across large distances.

Users perform a **soap interaction** by keeping the hull stationary and instead flipping the core (Figure 4c, also Figure 2). This may be considered the opposite of a belt interaction. Since this particular motion reminds us of the way one can spin a wet bar of soap in the hand, this interaction style inspired us to call the device soap. A soap interaction requires flipping the device over *twice*, so that the sensor ends up in its original location at the top of the device (using soap with the sensor in the back inverts the direction of the mouse cursor movement, i.e., either left and right or top and bottom directions are flipped).

Soap interactions produce very large motions and are particularly useful for getting across longer distances, e.g., while interacting with large screens.

SCENARIOS AND TEST APPLICATIONS

We have tried soap in the following four scenarios.

Controlling Microsoft Windows on wall displays

We have used soap to interact with the 18-panel 7680 x 3072 pixel wall display shown in Figure 5. Soap allowed us to do so while walking in front of the display, which was useful for reading details up-close.

The soap interaction style proved particularly useful for this application scenario as it allowed us to move the pointer in increments of one to six screens, depending on the acceleration settings. At the same time, joystick interaction allowed precise manipulation required for the acquisition of a window resize handle, for example.

Similarly, we used soap to control Windows Media Center™ while sitting on a sofa or walking across the living room.

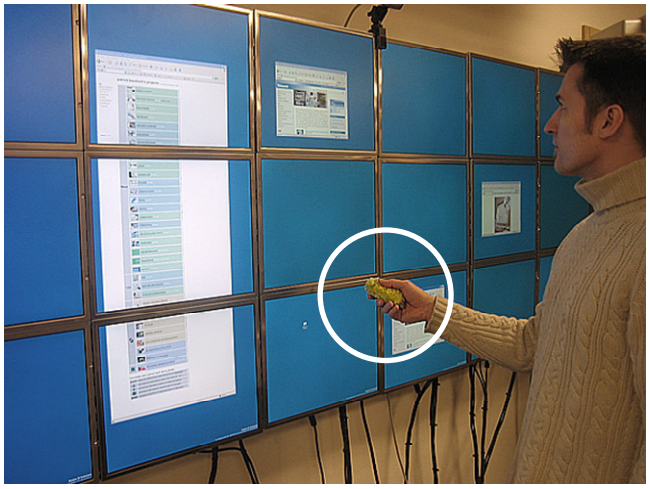

Figure 5: Using soap to interact with an 18-panel wall display.

Soap as a presenter tool

We used soap as a replacement for a wireless slide presenter. While traditional presenter tools use joysticks and are therefore rate-controlled, soap supports position input, which we found helpful for pointing. PowerPoint requires a click event for advancing slides, which soap provides when squeezed. PowerPoint requires a wiggle motion to invoke the menu, which was easily performed using a joystick interaction. Simple ink annotations such as underlining and circling were possible. However, more complex annotations such as scribbling were difficult.

Playing first person shooter games

The predominant input device setup for PC-based first person shooters is a keyboard and a mouse. The keyboard in the non-dominant hand is used for running and strafing; the mouse in the dominant hand controls the view. The latter includes aiming and shooting, which requires very fast and accurate control.

Replacing the mouse with soap and the keyboard with a wireless numeric keypad allowed us to move away from the table and made social settings, such as our sofa, more accessible. To make our numeric keypad "mobile", we turned it upside-down, glued 2 clips to the back, and then clipped it to a pant pocket, as shown in Figure 6a and b. This also allowed resting our hands on the keypad while playing.

While first person shooters require users to be able to continuously look and turn around, the user's vertical view is limited to the 180° range between straight up and straight down. This allowed us to use a specialized version of soap. This version had a longer body that prevented it from flipping over along the y-axis; instead, the device became self-centering along this dimension. Users could calibrate the vertical axis by performing a motion far enough to max it out; then they let it snap back (see Buxton's discussion of the nulling problem [2]).

We tested our input device combination with the game *Unreal Tournament 2004* (unrealtournament.com), which worked well. After about 30min of game play we beat the game on "novice" level; after 2 hours we beat it at "experienced", one level below our level for mouse-based gaming. We also tested the setup at a company-wide demo event. Figure 6b shows one of the attendees playing.

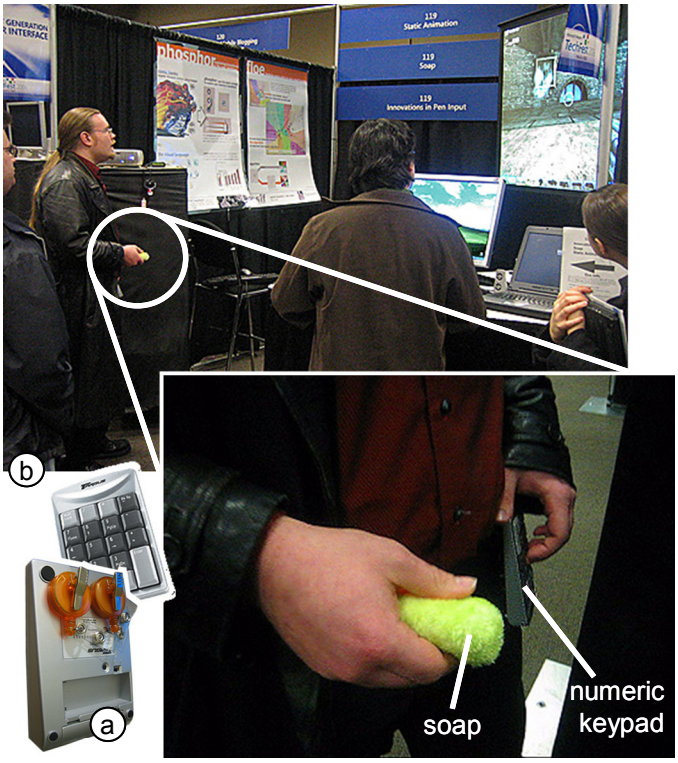

Figure 6: (a) Making the wireless keypad mobile. (b) A demo fest attendee playing Unreal Tournament 2004 using soap and a wireless keypad.

RELATED WORK

A variety of mid-air pointing devices have been proposed [3], some of which are available as products. The *Gyromouse* (gyration.com) uses a set of gyroscopes as sensors. Accelerometers detect tilt in *XWand* [11], *Tiltable interfaces* [9], and the *Nintendo Revolution* game controller (nintendo.com/revolution). Since gyro- and accelerometer-based devices offer no natural way of clutching, they are typically provided with an explicit clutch button [9].

Vision-based systems allows for a broad range of pointing devices, some of which are hands-free. In combination with a piece of paper, a vision system can be used to emulate mouse and joystick input [13]. GelForce senses directional force over each point of a rectangular area [10]. In Touchlight [12] the vision system detects optical flow.

One of the drawbacks of the devices listed above is that users can find it difficult to produce *no* input, as the device will perceive any jitter or tilt as input [6]. Input devices designed to rest on a table surface, in contrast, such as mice, trackballs, and track pads do not suffer from this limitation. The user's hand simultaneously touches the moving part of the device *and* the table/reference system, which provides haptic feedback and helps users perform accurate motion, even if the amplitude of the motion is small.

Some mid-air pointing devices offer the same benefit by using the non-dominant hand as a reference system [6]. The base of a joystick, for example, can be held by the non-dominant hand. In the case of TangiMap, the dominant hand holds a camera phone, while the non-dominant hand moves a bar code map [2]. Holding both the device and the reference system allows users to perform any type of motion without affecting input, as long as that motion affects both hands in synchrony. In particular, users can walk around while using these devices.

The *perific mouse* (perific.com) avoids the need for using both hands by putting device and reference system into the same hand, here a track ball that users can operate using a single hand. Thumb wheels, as found in some PDAs, offer the same functionality in one dimension. One way of looking at the perific mouse is to consider it a mechanical mouse that was adapted to one-handed mid-air use by turning it up-side-down.

Turning an optical mouse up-side-down and using bare fingers to operate it, unfortunately, does not lead to reliable tracking. In part, this problem inspires soap. By changing the core shape and adding the hull, the resulting device not only tracks reliably, but also offers several new interaction styles, as described earlier.

Soap is also related to tread-based controls, such as the Allison Research Slider [1] a variation of which is commercially available [8]. These devices are designed for stationary use, but we can create a mid-air version by replacing the hull of a soap device with a tread.

CONCLUSIONS

We presented soap, a mid-air pointing device based on an optical mouse. Soap shares many, but not all, of the strengths of its table-bound ancestor. First, soap is based on an optical sensor, which offers very high resolution sensing. Second, soap is a positioning device, not a rate-controlled device. Third, soap provides a tactile sensation when used, which helps users understand when they have moved and by what amount.

In addition, the *soap interaction* provides additional means for moving across long distances, which makes it an interesting candidate for large display application.

On the flipside, soap belt interaction is comparably slow and requires a more coordinated finger motion than, for example, clutching a mouse.

We are currently optimizing the hardware design of soap, addressing issues such as size, wireless range, and materials. We are also working on optimizing the emotional design of soap [7], as several people we observed seemed to enjoy performing belt and soap interactions independent of their original purpose. As part of the design process, we conducted a one day workshop where about 40 people constructed their own soap devices. As future work, we plan to assess the performance of soap using a Fitts' law study.

Acknowledgements

Many people have provided suggestions, materials, access to lab time etc. to this project, in particular Steve Bathiche, Kevin Flick, Hugh McLoone, Ken Hinckley, Jim Crawford, Thamer Abanami, Duane Molitor, Shirlene Lim, and John Lutian. Thanks also to Kim Young for the sketch of a soap mobile audio player used in the video.

REFERENCES

1. Buxton, W. & Myers, B. (1986). A study in two-handed input. *Proceedings of CHI '86*, pp. 321-326.
2. Buxton, W. There's More to Interaction than Meets the Eye: Some Issues in Manual Input. In Norman, D.A. and Draper, S.W. (eds.). *User Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates. Hillsdale, New Jersey. 1986.
3. Card, S., Mackinlay, J., and Robertson, G. A morphological analysis of the design space of input devices. In *TOIS* 9(2):99–122, April 1991.
4. Hachet, M., Pouderoux, J., Guitton, P. TangiMap—A Tangible Interface for Visualization of Large Documents on Handheld Computers. In *Proc. GI'05*, pp 9-15.
5. Jacob, R. Human-Computer Interaction: Input Devices. *ACM Computing Surveys*, vol. 28, no. 1, pp. 177-179 (March 1996).
6. Kohli, L. and Whitton, M. The Haptic Hand: Providing User Interface Feedback with the Non-Dominant Hand in Virtual Environments. In *Proc. GI 2005*, pp. 1-8.
7. McLoone, H. Touchable Objects. In *Proc. International Conference on Affective Human Factors Design* Asean Academic Press, London, 2001.
8. Penny and Giles Endless Belt www.pennyandgiles.com/products/products.asp?intElement=1174
9. Rekimoto, J., Tilting Operations for Small Screen Interfaces. In *Proc. UIST'96.* pp. 167–168
10. Vlack, K., Mizota, T., Kawakami, N., Kamiyama, K., Kajimoto, H., Tachi, S. GelForce: A Traction Field Tactile Interface. In *CHI'05 Extended Abstracts.*
11. Wilson, A. and S. Shafer. XWand: UI for Intelligent Spaces, In *Proc. CHI'03*, pp 545-552.
12. Wilson, A. TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction, *Proc. ICMI'04*, pp.69-76.
13. Zhang, Z., Wu, Y. Shan, Y., and Shafer, S. Visual Panel: Virtual Mouse, Keyboard and 3D Controller with an Ordinary Piece of Paper. In *Proc. ACM Workshop on Perceptive User Interfaces 2001*.