May 2000

WRL
Research Report 2000/3

Rock ’n’ Scroll is Here to Stay

Joel F. Bartlett

COMPAQ **Western Research Laboratory** 250 University Avenue Palo Alto, California 94301 USA

The Western Research Laboratory (WRL), located in Palo Alto, California, is part of Compaq's Corporate Research group. Our focus is research on information technology that is relevant to the technical strategy of the Corporation and has the potential to open new business opportunities. Research at WRL ranges from Web search engines to tools to optimize binary codes, from hardware and software mechanisms to support scalable shared memory paradigms to graphics VLSI ICs. As part of WRL tradition, we test our ideas by extensive software or hardware prototyping.

We publish the results of our work in a variety of journals, conferences, research reports and technical notes. This document is a research report. Research reports are normally accounts of completed research and may include material from earlier technical notes, conference papers, or magazine articles. We use technical notes for rapid distribution of technical material; usually this represents research in progress.

You can retrieve research reports and technical notes via the World Wide Web at:

http://www.research.compaq.com/wrl/

You can request research reports and technical notes from us by mailing your order to:

Technical Report Distribution
Compaq Western Research Laboratory
250 University Avenue
Palo Alto, CA 94301 U.S.A.

You can also request reports and notes via e-mail. For detailed instructions, put the word "Help" in the subject line of your message, and mail it to:

wrl-techreports@pa.dec.com

Rock ’n’ Scroll is Here to Stay

Joel F. Bartlett*

Abstract

Using the example of an electronic photo album, a novel user input method for digital appliances is introduced. Based on tilting and gesturing with the device, the Rock ’n’ Scroll input method is shown to be sufficient for scrolling, selection, and commanding an application without resorting to buttons, touch screens, spoken commands or other input methods. User experiments with a prototype system showed that users could quickly learn how to operate such devices and offered some insight into their expectations. These positive results, combined with improvements in the sensor technology, encouraged us to incorporate this input method into an experimental handheld system.

1. Introduction

Not missing a chance to show off the latest pictures of your children, you reach for your new photo album. As you remove it from your pocket, it activates and you see a display of photograph thumbnails in the album. Tilting the album on either axis scrolls through the thumbnails until you find the pictures you want to show. A gentle fanning gesture zooms in on the first picture, then you hand the album to your friend. After admiring the picture, she gestures to step through the rest of the album. The pictures are in both landscape and portrait mode, so a simple gesture is all that’s required to reorient the album to best display them.

Before putting it back in your pocket (where it will automatically shut down), you stop to admire the album itself. The album’s dimensions are that of the display with the addition of a thin, black border. In keeping with its spare, elegant design, it has no buttons or other visible controls: all functions can be accessed by direct manipulation.

While such an appliance has yet to reach the market, my colleagues and I have constructed a prototype that demonstrates the user interface we call Rock ’n’ Scroll.

The photo album is an example of a device with an Embodied User Interface[1], where the control mechanism is directly integrated into the album’s display. The design of such a device draws on a long-standing interest in using motion sensing for user input and the realization that inertial sensing systems are the logical type of system to embed in small devices[2]. While researchers have anticipated tilting a personal digital assistant (PDA) to navigate through a document, only in the last two to three years have the sensors become small, cheap, and sufficiently low power that they can be embedded in a handheld device and realize this vision.

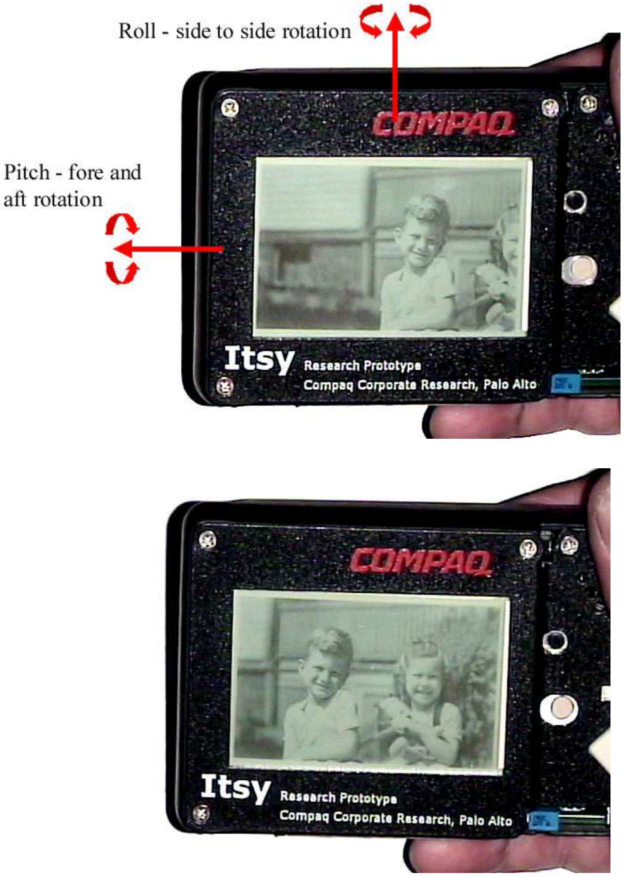
Roll - side to side rotation
Pitch - fore and aft rotation
COMPAQ
Itsy
Research Prototype
Compaq Corporate Research, Palo Alto
COMPAQ
Itsy
Research Prototype
Compaq Corporate Research, Palo Alto

Figure 1: As the left edge of the device is dipped, that is, rotated about the roll axis, the picture slides to the left of the display.

2. Tilt to scroll

For desktop devices, scrollbars have become the de facto control mechanism for paging through documents larger than the screen. When user interface designers use scrollbars on the smaller display of a handheld device, two problems arise: both horizontal and vertical scrollbars are often required, and the scrollbars occupy a larger percentage of the screen. Designers can recover the display area

---

* Compaq Computer Corporation Western Research Laboratory, 250 University Avenue, Palo Alto, CA 94301.
joel.bartlett@compaq.com.

This report is a superset of *Rock ’n’ Scroll is Here to Stay*, published in the May/June 2000 issue of *IEEE Computer Graphics and Applications*.

© 2000 IEEE Computer Society.
© 2000 Compaq Computer Corporation.

used by scrollbars by allowing scrolling by dragging the document using a stylus or scrolling via a cursor key. However, all these techniques still require the use of both hands. One way to free a hand is to use the hand that holds the device to control scrolling. As users rotate the device in their hand about either or both axes away from a user-defined neutral orientation, the device senses the motion and the screen contents scroll appropriately, as shown in Figure 1.

As described, scrolling is always on, just like dexterity puzzles that you tilt to get the rolling ball in the correct hole. This isn’t always the desired behavior, so other systems that have investigated “tilt to scroll ”have provided a “clutch ” button to engage scrolling[3-6]. When pressed, the device’s tilt scrolls the display; otherwise, scrolling is disabled.

The clutch button seems a good solution to accidental scrolling, but it comes at some cost. In some simple user tests that we conducted at the start of this investigation, users complained about having to tilt the device and press the button in a synchronized manner. For tasks like reading a column of text, users often want to continuously scroll and don’t want to keep their hand tensed to hold the button down the whole time.

Finally, any time you provide a button, you make an assumption about how the user will hold the device and push the button. One solution to this problem is to add more buttons. A more elegant one is to make the whole device a button, where the user squeezes it to enable scrolling[4]. Or, you can make the scrolling behavior modal.

Rock ’n ’Scroll uses the last approach. We assume scrolling is the device’s normal mode and provide commands to disable and enable scrolling and to set the device’s neutral orientation. However, for this behavior to be an improvement over a “clutch ”button, the application designer must ensure that frequent mode switches aren’t required.

3. Gesture to control

The commands that control scrolling can be issued in any number of ways, including button presses or interaction with a touchscreen. The same inertial sensors that control scrolling can also be used to recognize gestures as commands - this is the method we chose for Rock ’n’ Scroll. Levin and Yarin[7] have also investigated this method to implement a tilt-and shake-sensitive interface for drawing.

Other possible command gestures include snapping or shaking the device, tapping on it with the other hand, or fanning it. Unlike Levin and Yarin’s implementation[7], we rejected snapping and shaking the device because of the strain it places on the user’s hand and wrist. The mass of a 4-to 6-ounce device proves sufficient to strain users when they repeatedly perform such a gesture. Tapping is attractive because it doesn’t require users to move the device, but

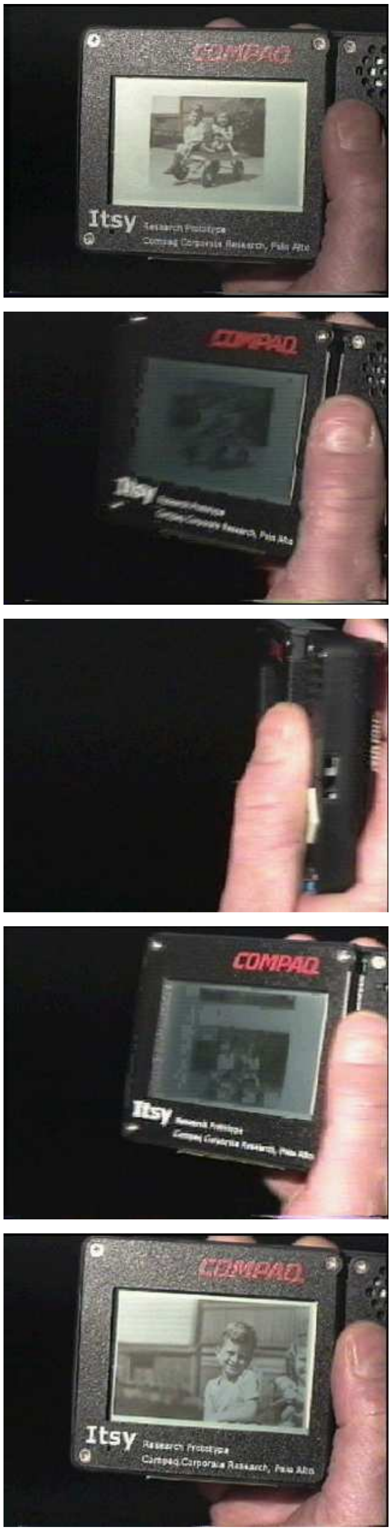

Figure 2: The first frame of the sequence (top-to-bottom) shows the thumbnail photo. In the next three frames, the user fans the device. The fifth frame displays the full-size picture.

it does have the disadvantage of requiring a second hand to operate.

We then tried slower fanning gestures about either of the device axes. The gentler motion seems easier on the hand and wrist than a snap, and the longer duration of the gesture makes it easier to separate it from high-frequency noise like hand tremors and vehicle vibrations. The sequence of pictures in Figure 2 illustrates how to select a picture from the album with a downward fanning gesture about the roll axis. The first frame shows the thumbnail picture. Then the user smoothly dips and raises the left-hand edge of the device to display the desired picture.

Three other similar gestures - upward about the roll axis, downward about the pitch axis, and upward about the pitch axis - can also be made. When combined with scrolling, they provide a vocabulary of commands more than sufficient to implement the rest of the album's commands: step from a picture back to the menu of thumbnails, step to the next picture, disable and re-enable scrolling, and reverse scrolling direction.

The gestures don't require a large amount of space to execute. For example, they can be performed with a user seated with an elbow on a chair's armrest. When users hold the device as shown in Figure 2, the device motion is in front of them, and those seated on either side aren't disturbed.

4. Hold still to orient

The final gesture used in the photo album is holding the device still for a couple of seconds, which serves two purposes. The first is that it reorients the device. In Figure 3, the user moves the album from landscape to portrait mode by positioning it vertically with the new top edge of the album up. The user holds the album still in that position for a couple of seconds, then the screen image rotates to the new orientation.

The second purpose of the hold-still gesture is to change the album's neutral orientation. To do this, the user first gestures to disable scrolling, then holds the device still in the new neutral orientation for a couple of seconds, and gestures to enable scrolling. While this operation takes three gestures to accomplish, it's not a burden to users because they perform the task infrequently.

5. Implementation

Now that I've demonstrated the mechanics of controlling the device, we can turn our attention towards its implementation. Verplaetse[2] characterized the motion that should be expected as accelerations from $0.1g$ to $1.0g$ and a frequency of motion of 8 to 12 Hz. The inertial detectors must also meet the needs of a personal device: rugged, self-contained, small, light, and low power. Finally, we need low-cost detectors to meet the manufacturing cost requirements for mass-market digital appliances.

With these design constraints in mind we chose a two-axis, single-chip accelerometer - Analog Devices' ADXL202 - to measure acceleration on each of the device's axes (see Figure 1). With the addition of one resistor and three capacitors, we integrated it into Itsy[8], Compaq Research's experimental platform for “off the desktop ”computing. Every 10 ms, the ADXL202 reports acceleration values in the range of $-2g$ to $+2g$ for each axis. These measurements are averaged in groups of four, so the photo album application sees 25 sets of measurements per second. These values represent a combination of gravitational acceleration, user motion, and noise. Initial noise filtering occurs by using a low-pass filter in the accelerometer set at 10 Hz. Exponentially averaging sequential results achieves additional filtering. When plotted over time, the values look something like Figure 4. At time *t0* the user is scrolling the display by rocking it fore and aft. By time *t1*, the user is holding the device fairly still. At time *t2*, a gesture like that shown in Figure 2 starts and is completed at time *t3*.

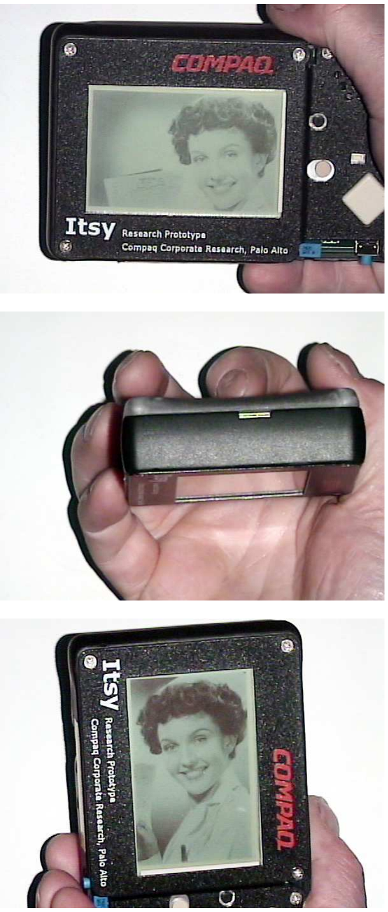

Figure 3: As shown in this sequence, the user positions the album in the desired orientation, then holds it for a couple of seconds to change the display from landscape to portrait mode.

Figure 4 illustrates a problem with extracting both gesture and scrolling information from the same data stream. The user thinks the gesture starts at time *t2*, but the device doesn't recognize it until *t3*. During this interval, scrolling continues on both axes, so the application may scroll from one item to the next in a menu and the gesture's action is applied to the wrong object. This can be seen in Figure 2, where in the fourth frame, the thumbnail menu is scrolling because the gesture has not yet been recognized. To correctly associate the gesture with the right device state, the scrolling code is speculatively executed by saving the time and current scrolling state before each change occurs. When the system recognizes a gesture at *t3*, the system restores itself to the state at *t2* (shown in the first frame of Figure 2), then applies the gesture's operation. The scrolling state need not be saved indefinitely, since a maximum length of a gesture can be defined.

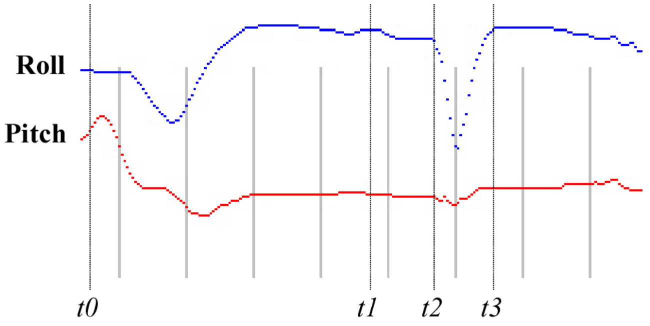

Figure 4: Acceleration versus time for each axis of motion is plotted here. Each signal is labeled and shown independently, with the acceleration value on the vertical axis and time on the horizontal axis. The vertical gray bars are at one-second intervals, and dotted vertical lines denote points of interest.

6. But will it work?

Before implementing Rock 'n' Scroll for Itsy, we constructed a prototype system to test the ideas. The prototype system appears to the user as a handheld computer with a cable leading out of it. It consists of a small video liquid crystal display (LCD) with a two-axis accelerometer attached to the back. The accelerometer was made from two earlier generation (Analog Devices ADXL05) one-axis accelerometers and included a PIC16C73A microcontroller to convert the analog outputs from the ADXL05s into digital values reported 25 times a second to a PC via an RS-232 serial line. The measurements drive a PC application whose output to the screen is converted to a National Television System Committee (NTSC) video signal via a scan converter, then displayed on the LCD. The prototype is decidedly inferior in handling to an Itsy, since it is larger and heavier, its screen has a narrower viewing angle, and it is tethered to a PC.

Applications constructed include a viewer allowing users to scroll through a number of pictures and text images, a game where a cat catches mice, and an interactive map of our building. The map application lets users tilt to browse a floor and gesture to change floors, set the device's neutral orientation, or disable or enable scrolling. Fifteen members of Compaq's Palo Alto research labs participated in a scripted trial that took about twenty minutes per person. During the trial the user stood and manipulated the test system under my instruction.

The most important goal of the study as to confirm (or reject) the premise that the Rock 'n' Scroll user interface can be rapidly mastered by a larger user community. Test users first familiarized themselves with the device by scrolling text and pictures. With little practice, they were able to start and stop scrolling, find specific articles in the text, and center selected items in a picture on the screen. Next, we demonstrated the map application, then the users tried it. A subjective measure of their ability to operate the application as then made based on the three to five minutes that they used the application. Three quarters of the users found it easy to operate all features of the application. One quarter of the users had moderate difficulty making one or more of the types of gestures. With some coaching and adjusting of the gesture recognizer, all users were able to make all types of gestures.

The more interesting result though, has to do with user expectations about how the device would scroll. As the prototype hardware and software were being constructed, we observed that users have different (and often strong expectations) about which way the display should scroll when they tilted the device. For example, when the left edge of the device is dipped, should the picture on the device slide left or right? We designed the user study to examine this and found that while a default sensitivity and scrolling speed was acceptable, a default scrolling direction as not. For each user, the initial scrolling behavior as randomly selected. As the tests were run, the users were allowed to change scrolling behavior to a preferred setting, while the experimenter encouraged them to try the alternative as well. All but two test subjects immediately chose a preferred direction and maintained it across the entire test. The results, summarized in Figure 5, suggest that the world is divided into "sloshers" and "pointers." Sloshers expect that the screen contents have mass and tilting the display will cause them to slide down. Pointers expect the opposite motion as they point with the edge of the device towards the section of the image that they want moved to the center of the display.

However, when presented with a simple game, all subjects but one expressed a strong preference for the "slosh" mode of operation. The game, shown in Figure 6, displayed a cat on the screen. The user tilted the device to move the cat around to catch mice. Those who favored the "slosh "mode of operation in the earlier parts of the test explained that when they dipped the left edge of the device, they expected the cat to go left. Those who "point" explained that they were pointing to where they wanted the cat to go. Though the test subjects had different explanations, the end result as that they expected the device to behave in the same manner.

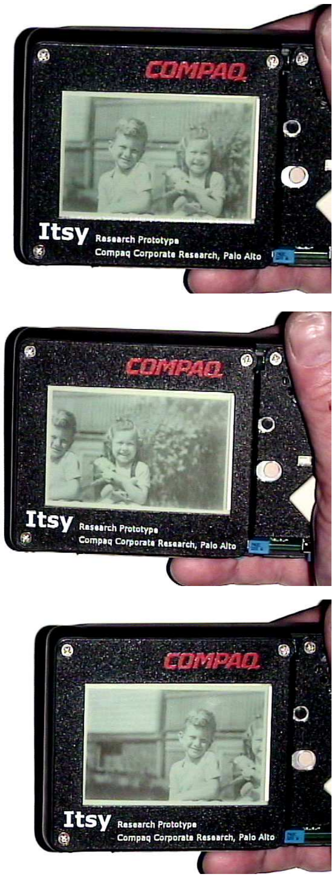

Figure 5: In the frame at the top, the picture is in the initial position. When the left edge of the device is dipped, seven of the fifteen subjects expected gravity to "slosh" the picture into the position it has in the center frame, and five of the fifteen subjects expected to see the picture like the bottom frame because they "pointed" with the left edge of the device to that corner of the picture.

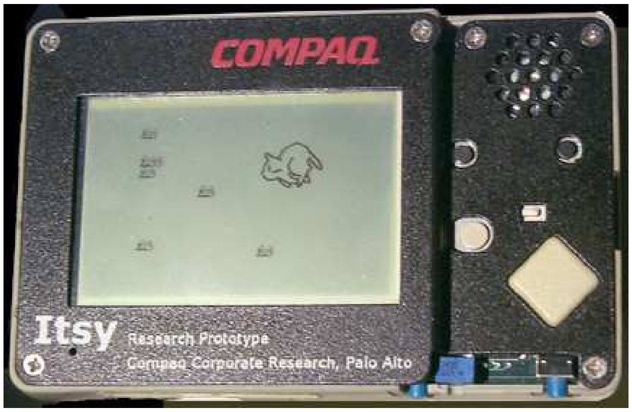

Figure 6: Cat and mouse game.

We also observed this behavior with the game Doom that we ported to Itsy. In Figure 7, virtually all users expected that when they tilted the top edge of the display down, they'd move up the stairs. Whether they sloshed the gun up the stairs or pointed to where they wanted the gun to go, the action and expected result were identical.

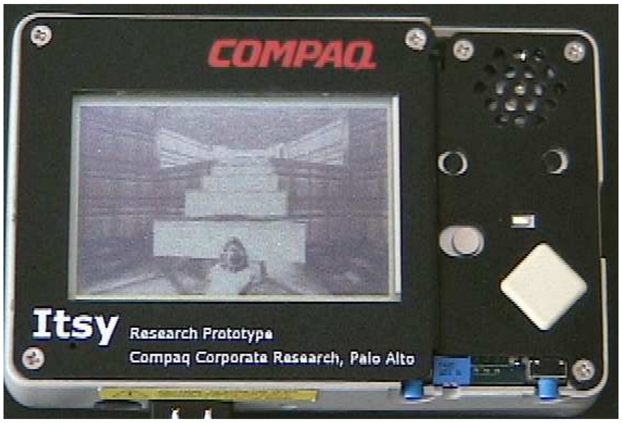

Figure 7: Doom on Itsy

Directly manipulating a handheld computer to play games fascinated nearly all subjects. The simple cat and mouse game on the prototype as surprisingly compelling, and users would often continue to play with it for a few minutes after their test session ended. When watching users play Doom on an Itsy, it's clear that they had significantly more kinesthetic satisfaction in tilting the device than in rocking a cursor key.

7. A limit to embodied user interfaces

An alternative way to play Doom is to maneuver using the diamond-shaped cursor key to the right of the screen. This offered an opportunity to determine user input preferences for this application. We encouraged users to play the game using both input methods and polled them later about their preference. The seven respondents to the poll were evenly split in their preferences: three preferred the cursor key and four preferred tilt-to-scroll. However, when their user interface preference was correlated with their level of experience playing electronic games, all those with extensive experience preferred to use the cursor key and the rest preferred tilt-to-scroll. Experienced users felt they could maneuver with more precision with the cursor key and didn't like the fact that an enthusiastic press of the fire button would move the player. Less experienced users preferred to tilt-to-scroll and described that method as more natural or intuitive than the cursor key.

While most users had positive things to say about the Rock 'n' Scroll interface, they also noted some problems with the display while playing the game. The Itsy display is a reflective LCD, so ambient light conditions can cause significant changes in screen visibility as the device tilts. In addition, enthusiastic play often results in significant tilt that exceeds the viewing angle of the display. While LCD technology will improve and reduce these effects, some experienced users observed one display difficulty inherent in embodied user interfaces: perceived motion on the screen is the sum of the motion of the embodied device and the changes made to the display by the device. As you interact with the device by moving it, the orientation of the display to the user changes, then in response to that motion the display contents move on the display. For "twitch "games like Doom, where the user manipulates the device rapidly, tilt-to-scroll results in motion on the screen that may be harder to visually track than motion via the cursor key. On the other hand, for applications like the photo album, where the user is not rapidly manipulating the device while trying to track moving fine detail on the screen, this should not be an issue. The results from this small test suggest an area for further study.

8. Additional ways to Rock 'n' Scroll

In the discussion to this point, users always held the device in either or both hands. However, since all interaction occurs by manipulating the device, it could also be worn. One promising area for wearing a small gesture-controlled device is on the "snuff box "portion of either hand - that is, the area between the thumb and index finger on the back of the hand. This area is typically visible when using your hands. Gestures can be made on either of the device's axes by moving either your hand or forearm. Figure 8 shows how a thermometer that measures the temperature at the palm of the glove could be mounted. By gesturing, the user can change the display to show the minimum, maximum, or current temperature. Other gestures could be used to switch the scale between Fahrenheit and centigrade and clear the saved minimum and maximum values. While at first this might seem a contrived example, consider for a moment the difficulty of trying to push your watch's buttons while you have gloves on.

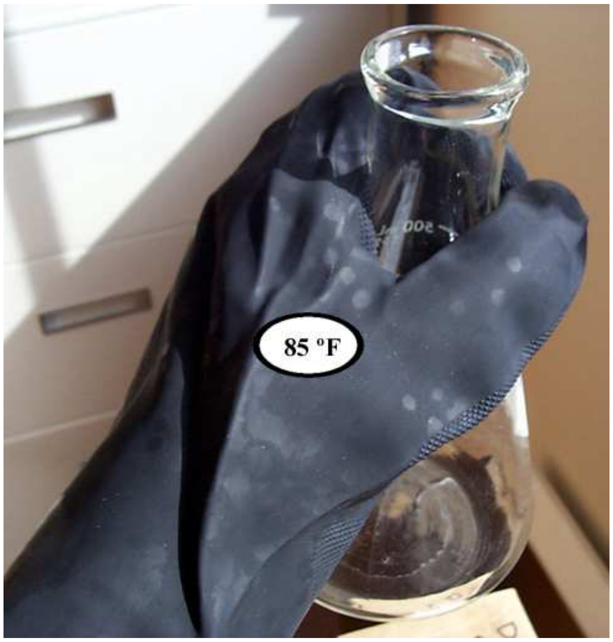

Figure 8: Industrial glove with hands-free thermometer (mock-up).

To date this work has focused on interacting with a handheld device with a display, but the same techniques could be used to control other personal electronics such as radios, MP3 players, or cellular telephones. Rather than being restricted to play, fast-forward, and reverse buttons on a music player, the user could tilt to have fine-grain control over playing speed and gesture to step to the next track. As fashion plays an increasing role in personal electronics, invisible control interfaces could provide desired product differentiation. Finally, the inertial sensor doesn't have to be incorporated in the device, but could be in the form of a ring or bracelet that communicates with the device via a personal wireless network.

9. Evolution of sensors

Moore's Law predicts continuing improvements in digital electronics, equally impressive improvements are also occurring in sensors. Three generations of Analog Devices' accelerometers were tracked during the design and construction of Rock 'n' Scroll and their key attributes are shown in the following table.

	ADXL05	**ADXL202**	**ADXL202E**
axes	1	2	2
output	analog	digital	digital
voltage	4.75-5.25	3.0-5.25	2.7-5.25
current	10 mA	0.6 mA	0.6 mA
height	5.8 mm	5.5 mm	2 mm
area	70 sq mm	110 sq mm	25 sq mm

Our investigation started with the first generation part, the ADXL05. Since each part measured only 1 axis, two were required for the design, as well as a microcontroller to convert the analog outputs to appropriate digital values, and a power regulator to convert Itsy's 3.3 volt supply to 5 volts. Prototype electronics were constructed, but the number of components and the power required (~25 mA) suggested that the design was not appropriate for Itsy.

The next generation part improved things significantly. Since the ADXL202 measured 2 axes, only one was required. Its digital outputs eliminated the microcontroller and provided a further reduction in components, but the big improvement that enabled Rock 'n' Scroll's integration into Itsy was the 40x reduction in power: 0.6 mA versus ~25 mA. Package height and area continued to be a problem though, and resulted in Itsy's length growing by 10 mm.

The final generation of the part (which was too late for our use), the ADXL202E, significantly reduces the package height and area which will simplify its integration into future digital appliances.

10. Conclusion

The Rock 'n' Scroll user interface shows how inertial sensors in handheld devices can provide additional function beyond "tilt-to-scroll." By also using them to recognize gestures, a significantly richer vocabulary for controlling the device is available that implements an electronic photo album, pager, or other limited function digital appliance without any additional input methods. The examples in this article offer a glimpse at the freedom for both the device designers and users inherent in devices that can be held in either hand, at any orientation, operated with mittens on, or not in the hand at all.

I hope I've communicated some of the excitement that surrounds inertial interfaces. What started out as a side investigation in our laboratory has captured sufficient interest that we've designed it into the next generation of Itsy. We expect others to adopt these interfaces as well and look forward to seeing the results of their efforts.

Rock 'n' Scroll - the movie

A short movie demonstrating the sample photo album is available from the Rock 'n' Scroll project web page: http://www.research.compaq.com/wrl/projects/RocknScroll/RocknS.html.

Acknowledgements

Rock 'n' Scroll as developed as part of the Itsy pocket computer project at Compaq 's research labs in Palo Alto. The work relied on the interest and knowledge of other team members including William Hamburgen, Lawrence Brakmo, Keith Farkas, Dirk Grunwald, Tim Mann, Robert Mayo, Sharon Perl, Barton Sano, Marc Viredaz, Carl Waldspurger, and Deborah Wallach. Wayne Mack provided electronics assembly assistance. My colleagues at Compaq's research labs in Palo Alto were willing and articulate test subjects. Wendy Bartlett and CG&A's editors, guest editors and reviewers improved the presentation of the results. I thank you all.

References

[1] K. P. Fishkin, T. P. Moran, and B. L. Harrison, "Embodied User Interfaces: Towards Invisible User Interfaces," Proc. Engineering for Human-Computer Interaction (EHCI 98), S. Chatty and P. Dean, eds., Kluwer Academic Publishers, Hingham, Mass., 1998, pp. 1-18.

[2] C. Verplaetse, "Inertial Propriceptive Devices: Self-Motion-Sensing Toys and Tools," IBM Systems J., Vol. 35, No. 4/5, 1996, pp. 639-650.

[3] G. W. Fitzmaurice, "Situated Information Spaces and Spatially Aware Palmtop Computers," Comm. ACM, Vol. 36, No. 7, July 1993, pp. 38-49.

[4] B. L. Harrison et al., "Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces", Proc. Human Factors in Computing Systems (CHI98), ACM, New York, 1998, pp.17-24.

[5] J. Rekimoto, "Tilting Operation for Small Screen Interfaces, "Proc. ACM Symp. User Interface Software and Technology (UIST96), ACM, New York, 1996, pp.167-168.

[6] D. Small and H. Ishii, "Design of Spatially Aware Graspable Displays", Proc. CHI97, ACM, New York, 1997, pp. 367-368.

[7] G. Levin and P. Yarin, "Bringing Sketching Tools to Keychain Computers with an Acceleration-Based Interface", Proc. CHI98 Extended Abstracts , ACM, New York, 1998, pp. 268-269.

[8] C. Waldspurger, "The Itsy Pocket Computer," invited talk, Int'l Symp. Wearable Computers 98, http://www.research.compaq.com/wrl/itsy/talk-iswc98/.