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.

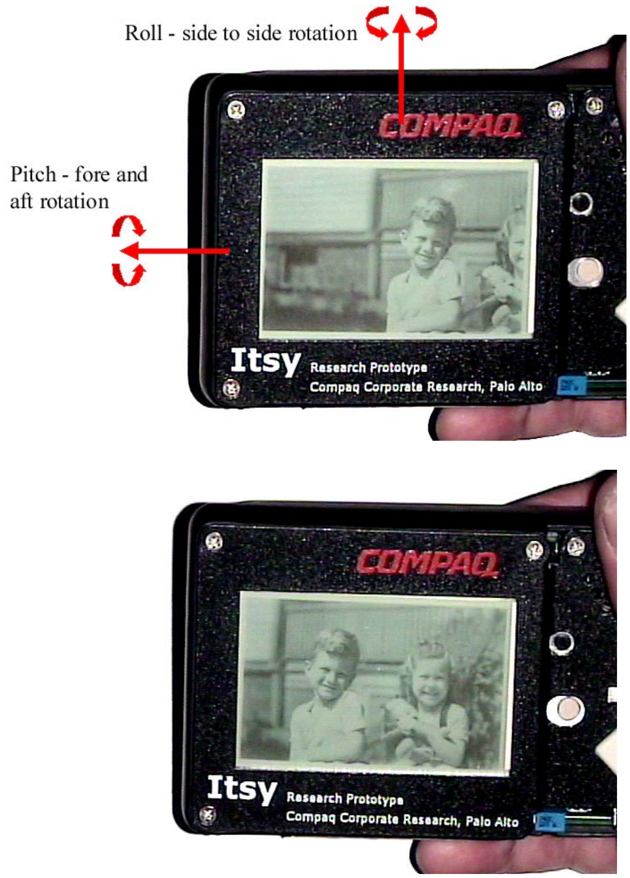

Figure 1: As the left edge of the device is dipped, that is, rotated about the roll axis, the picture slides to the left of the display.

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*.

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 0.1*g* to 1.0*g* 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'

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.

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 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,

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.